AFIT/GCS/ENG/92-20

EFFICIENT DERIVATION AND APPROXIMATIONS OF CEPSTRAL COEFFICIENTS FOR SPEECH CODING

THESIS

Kimberly Ann Limcangco
Captain, USAF

AFIT/GCS/ENG/92-20

'Approved for public release; distribution unlimited'

AFIT/GCS/ENG/92-20

EFFICIENT DERIVATION AND APPROXIMATIONS OF CEPSTRAL COEFFICIENTS FOR SPEECH CODING

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Kimberly Ann Limcangco, B.S.

Captain, USAF

December, 1992

Acknowledgments

I thank my advisor, Dr. Bruce Suter, for his guidance and assistance in my thesis effort. I also thank my committee members, Major Greg Warhola and Dr. Tim Anderson, for their suggestions and contributions.

I express my sincere appreciation to Dr. Robert McAulay of MIT Lincoln Laboratories for his expert guidance and assistance in pursuing the theory behind this research effort, and for allowing me the opportunity to conduct this research.

I must thank my parents, John & Hazel Walther, for their encouragement.

Finally. this thesis would not have been possible without the support of my husband Michael. His love and support made all the difference!

Kimberly Ann Limcangco

Table of Contents

		Page
Acknowledgments		ii
Table of Contents		iii
Abstract		vi
I.	Introduction.	1-1
	Background	1-1
	Problem Statement	1-3
	Research Objectives	1-3
	Research Questions	1-4
	Definitions	1-4
	Cepstrum.	1-4
	Frequency Sampling.	1-5
	Frequency Warping.	1-6
	Real-time Processing.	1-6
	Assumptions	1-8
	Scope	1-8
	Summary of Presentation	1-9
II.	Background (Literature Review).	2-1
	Introduction	2-1
	The Speech Signal	2-2
	The Sinusoidal Transform Coder	2-3
	Analysis System of STC	2-4
	Pick peaking routine: Obtaining the Amplitudes and Frequencies of the Underlying Sine Waves.	2-4

Page

The SEEVOC Technique: an Alternative Method for Locating the Underlying Sine Waves. 2-5

Current Computation of the Cepstral Coefficients withing STC. 2-6

Parameters passed to Synthesis System 2-7

Synthesis System of STC 2-8

Block Diagram of STC 2-9

III. Methodology. 3-1

Introduction 3-1

Theoretical Background 3-1

Minima vs. Maxima 3-5

Application to Speech Coding 3-7

Solving $Bc = \gamma$ 3-11

Approximations for Real-time Environment 3-12

Tridiagonal Matrix Approximation 3-15

Identity Matrix Approximation 3-19

Toeplitz Matrix Approximation 3-19

Computing the Cepstral Coefficients Based on a Warped Spectral Envelope 3-23

Equipment and Support 3-24

Validation of Method 3-25

Summary 3-25

IV. Findings. 4-1

Findings for Research Questions 4-1

Quantifying Results 4-2

Computing the Number of Cepstral Coefficients 4-4

Results 4-8

The Exact System. 4-8

Toeplitz Approximations. 4-12

Page

Diagonal Approximations 4-16

Unvoiced Speech. 4-23

Summary 4-25

V. Conclusions and Recommendations. 5-1

Research Question One Conclusions 5-1

Research Question Two Conclusions 5-1

Research Question Three Conclusions 5-2

Summary 5-2

Appendix A. A-1

Bibliography BIB-1

Vita VITA-1

Abstract

A new formulation is presented for the calculation of cepstral coefficients directly from measured sine wave amplitudes and frequencies of speech waveforms. Approximations to these cepstral coefficients are shown to be suitable for operation in a real-time speech coding environment. These results were encoded in the *C* programming language and then evaluated through experiments that were conducted on the McAulay-Quatieri Sinusoidal Transform Coder (STC).

EFFICIENT DERIVATION AND APPROXIMATIONS OF CEPSTRAL COEFFICIENTS FOR SPEECH CODING

I. Introduction.

Background

The background for this thesis is the human speech process and the importance of speech processing applications to the United States military.

Human speech is produced by excitation of the vocal tract, which is an acoustic tube that runs between the lips and the glottis (14:723). In signal processing applications, speech is modeled by the following linear set of filters representing these characteristics (13:167):

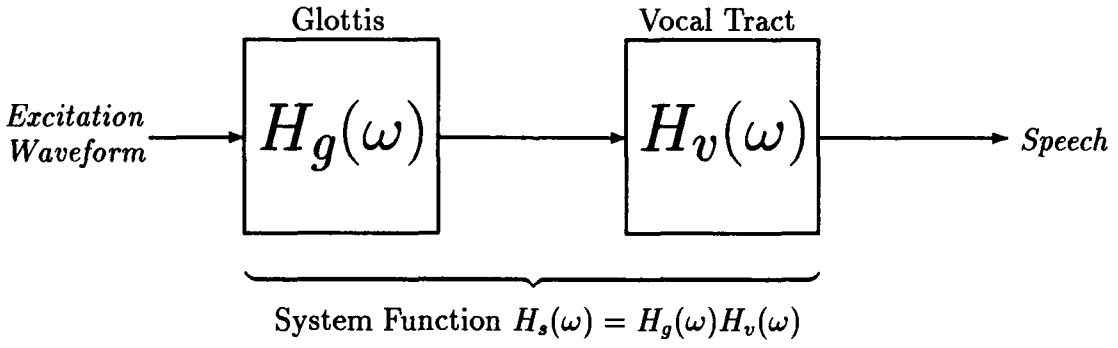


Figure 1.1. Speech Filter Bank.

Speech is a nonstationary signal that originates from the glottis, which produces either a quasi-periodic pulse train of airflow or a steady stream of airflow (the latter evolves into a noiselike excitation when the vocal tract is constricted). From the glottis, the airflow moves to the vocal

tract which imposes information on the glottal output through gestures of the mouth and lips, resulting in speech (14:724) (16:66).

Military applications which benefit from speech processing include security (speaker identification and access control), data transmission, and narrow-band communications. Perhaps the most well-known and wide-spread speech processing application within the military is the STU-III secure telephone unit, which uses a 2400bps speech coding algorithm to code and encrypt speech for transmission across a telephone line. As computer networks based on packet-switching technology become prevalent within the military, speech processing applications will be used for packet voice communications, advanced intelligent terminals, and voice control of resources and services (22:1627-29).

Speech processing areas of interest to the military include speech recognition, speech signal analysis, and speech coding. This thesis will concentrate on speech coding: the representation of the output of the human vocal tract in a digital form, and only one of many digital communications systems essential for military operations. As the requirement for digital communications is ever increasing, speech and other digital signals must be coded in as efficient form as possible. This is the impetus behind speech coding: the desire to significantly reduce the storage requirement for digitized speech signals while still maintaining the quality of the speech. Speech coding is therefore sometimes referred to as speech compression.

In some speech coding applications, it is desirable only to maintain the intelligibility of the speech. This is usually true in the case of "canned" recorded messages, like those used in telephone systems ("This number is no longer in service", etc). In other applications, however, is it desirable not only to understand what is being said, but to be able to recognize the speaker as well, such as in the transmission of digital speech over a toll-quality telephone line: the user not only wants to understand what is being said, but to recognize that Aunt Martha (or whoever) is the speaker.

This is especially important in secure voice systems: the user likes the extra assurance of being able to recognize a speaker's voice in order to confirm their identity.

Speech coding is influenced by three main factors: the intelligibility and quality of the encoded speech, the bit rate at which the speech is encoded (the amount of storage bits required for each second of encoded speech), and the computational complexity of the speech coder. It is desired to maximize the quality of the coded speech while maintaining a low bit rate and a limited computational complexity within the speech coding algorithm (16:225).

To a large extent, the computational complexity of the speech coder depends upon the algorithm selected. Two main classes of algorithms exist today: those that attempt to reproduce the original shape of the speech waveform and those that attempt to reproduce the sound of the original speech without attempting to keep the original waveform shape. The former are referred to as "waveform encoders" and the latter as "vocoders" (short for voice encoders). This thesis concentrates on a vocoder developed at MIT Lincoln Laboratories whose algorithm is based on a sinusoidal model for speech–the representation of the speech waveform as the summation of sine waves of various amplitudes, frequencies, and phases.

Problem Statement

Based on the application of the sinusoidal model of speech to speech coding, consider an approach to obtain cepstral coefficients directly from the measured sine wave amplitudes and frequencies of digitized speech waveforms.

Research Objectives

Research objectives under this thesis are to study, derive, and implement alternative algorithms to obtain an effective method for computing cepstral coefficients directly from the measured sine wave amplitudes and frequencies of digitized speech waveforms.

Research Questions

1. Can a correct algorithm be derived for a direct solution of the cepstral coefficients based on fitting a cepstral model to the measured speech data?
2. If so, what mathematical approximations to the algorithm may be derived? Which are the fastest and most efficient algorithms, to enable execution within a real-time environment?
3. What are the results? Does the algorithm or any of the approximations yield reconstructed speech perceptually equivalent to the original speech?

Definitions

Cepstrum. Cepstrum analysis is a nonlinear signal processing technique which has seen much success in processing signals such as speech signals, seismic signals, biomedical signals and sonar signals.

The term cepstrum was coined by J.R. Tukey, and is a play on the word "spectrum", hinting that the cepstrum is obtained by performing a further spectral analysis on the frequency spectrum.

Work under this thesis is based on A.V. Oppenheim and R.W. Schafer's definition of the complex cepstrum of a signal, found in (14:770). Here, the complex cepstrum is defined as the inverse Fourier transform of the logarithm of the system function. Considering the system function of Figure 1.1, and using the inverse Fourier transform integral in (14:46), the complex cepstrum of the system function, $H_s(\omega)$, may be written as:

$$c_m = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log H_s(\omega) e^{j\omega m} d\omega \tag{1.1}$$

where c_m are the cepstral coefficients and $j = \sqrt{-1}$. This equation is known as the cepstral envelope equation and is the basis for the cepstral analysis completed in this thesis.

Frequency Sampling. The frequency content of a signal is measured in hertz (Hz), which is the number of cycles per second. An intuitive grasp of frequency can be had by considering the following: A signal whose amplitude varies rapidly between positive and negative values relative to some average value over a short period of time is a "high frequency" signal. Likewise, a signal whose amplitude does not vary rapidly with time (stays relatively constant over a short period of time) is a "low frequency" signal.

The sampling frequency (or sampling rate) of a digital speech signal is the rate at which the analog (continuous) version of the speech signal is sampled. The sampling period, denoted by Δ, is the time duration between which samples are taken, and is the reciprocal of the sampling rate. If, for instance, a sample of the analog signal is taken every .1 seconds ($\Delta = .1$), the sampling rate is the reciprocal of this period ($\frac{1}{\Delta}$), or 10 samples per second. The sampling rate of a signal is also measured in hertz (Hz), and is the number of samples taken per second.

An important theorem which applies to the sampling of analog signals is the Nyquist Sampling Theorem. This theorem states that for any sampling period, Δ, there is a particular frequency, f_c, known as the *Nyquist critical frequency*, given by

$$f_c = \frac{1}{2\Delta}.$$

If the frequency content of a continuous signal is limited to frequencies less than or equal to f_c, then the Nyquist Sampling Theorem states that the continuous signal can be completely determined from a sampled version of itself with a sampling rate greater than or equal to f_c (18:403).

In order to choose an appropriate sampling rate for a particular analog signal, prior knowledge of the signal's frequency content must therefore be known. For speech signals, the major frequency components fall below 3000 Hz (19:23). Therefore, in general, a sampling rate greater than twice this frequency is sufficient for sampling the majority of speech signals. For research conducted under this thesis, speech signals will be sampled at a rate of 8000 Hz (8 kHz).

Frequency Warping. Since the human ear is less sensitive to higher frequency sounds (sounds with a higher pitch), a process called frequency warping, also known as spectral warping, is often used in speech coding. Frequency warping takes advantage of the human ear's lower sensitivity to higher frequencies by warping the spectral envelope onto a smaller scale, known most often as the *mel* scale.

When a discrete Fourier transform operation is done on a portion of a digitized speech signal, the resulting spectral envelope is by definition half the length of the discrete Fourier transform (DFT). For instance, a 1024-point DFT results in a 512-point spectral envelope, corresponding to "normalized" digital frequencies 0 to π. This spectral envelope can itself be sampled in a nonlinear fashion, so on the resulting warped scale, the number of points of the warped spectral envelope is less than that of the original spectral envelope.

When warping a spectral envelope, the warped envelope is equivalent to the linear envelope up until a determined frequency. From that point on, samples of the linear envelope are taken less and less often, which in effect deletes higher frequency details from the spectral envelope. Since the human ear is less sensitive to higher frequencies, a warped frequency envelope better matches the sensitivity of the human ear, and offers the advantage that there is less spectral information to preserve during the speech coding process.

Real-time Processing. A real-time process can be defined as a process that is accomplished without creating a delay noticeable to the user (6:1143). Speech coding is frequently a real-time process. For instance, during a secure telephone call, the coding and encryption of speech for secure transmission across a telephone line, to be decoded on the receiving end, is a real-time process: there should not be a noticeable delay between the time of the actual utterance and its deliverance to the user.

Since speech coding, along with other digital signal processing (DSP) applications, involves mathematically intensive algorithms, special computer hardware is used to facilitate the mathe-

matical computations. Today, this hardware is typically contained on a single microcomputer chip and is commonly referred to as a digital signal processor (DSP chip).

Even with the availability of the DSP chip, the real-time DSP software must be developed as efficiently as possible: inefficient software will not execute as fast as efficient software. The 90/10 rule of computer science states that 90% of the execution time of a computer program is spent on the execution of 10% of the code. Since, within STC, cepstral coefficients are computed as often as every 10 milliseconds, it is a reasonable assumption that for real-time operation, the cepstral coefficients should be computed as efficiently as possible. This is the goal of this thesis research: to explore and implement solutions for cepstral coefficients which are efficient and suitable for real-time processing.

The efficiency of the computation of the cepstral coefficients will be measured by using the asymptotic notation, known as O-notation, to describe the order of growth of an algorithm's running time. Here, the running time or computational complexity of an algorithm refers to the number of operations executed.

The operations that will be counted in computing the computational complexity are known as floating point operations (FLOPS). The concept of counting the number of FLOPS came into being to quantify the work that occurs during computer program execution. C.B. Moler defined a FLOP as the amount of work associated in executing the statment

$$s = s + a_{ik}b_{kj}$$

which includes the effort of doing a floating point add, a floating point multiply, and some subscripting (4:52).

As an example, consider the following algorithmic structure:

```
for (i=1; i ≤ N; i++){
    for (j=1; j ≤ N; j++){
        statement 1;
        statement 2;}}
```

where statement 1 and statement 2 are FLOPS. If statement 1 takes a constant c_1 time to execute and statement 2 takes a constant c_2 time to execute during each iteration, the order of growth of the algorithm could be expressed as $c_1N^2 + c_2N^2$ and the computational complexity would be written as $O((c_1 + c_2)N^2)$. As N grows large, the constant term becomes less significant in determining the computational efficiency of an algorithm. Therefore, the constant is frequently ignored and the computational complexity for this case may be denoted as $O(N^2)$. A detailed review of various asymptotic notations and the analysis of algorithms is found in (2).

Clearly, an algorithm with complexity $O(N)$ is more efficient than one with complexity $O(N^2)$, which in turn is more efficient than an algorithm with complexity $O(N^3)$. These are the three basic complexities which will be encountered in the algorithms used in this thesis effort. The algorithm found to yield the best results with the lowest computational complexity will be the algorithm most favored for real-time implementation within a multi-purpose speech coding environment.

Assumptions

1. Work will be consistent with currently established sinusoidal transform coding techniques developed at MIT Lincoln Laboratories.
2. Final implementation will be accomplished in the *C* programming language.

Scope

The work under this thesis includes a mathematical analysis of the derivation of cepstral coefficients from the measured sine wave amplitudes and frequencies of speech waveforms, and

the implementation of this analysis into the *C* programming language. This implementation will be suitable for compilation into a real-time speech coding system. Various algorithms will be considered and computational complexity analysis will be accomplished on these algorithms.

Summary of Presentation

Chapter 1 is an introduction to the thesis. Chapter 2 gives an overview of the current literature pertaining to the speech coder used in this thesis work, and highlights the underlying speech coding algorithm. Chapter 3 presents the methodology used to solve the problem stated in Section 1.2. The solution results are detailed in Chapter 4, and Chapter 5 concludes the thesis work.

II. Background (Literature Review).

Introduction

The speech coder on which the results of this thesis work are accomplished is the Sinusoidal Transform Coder, developed by Dr. Robert J. McAulay and Dr. Thomas Quatieri of Massachusetts Institute of Technology (MIT) Lincoln Laboratories. The development of STC, its enhancements and modifications have been published regularly by IEEE (8-13) and other various publications.

The STC falls into the broad class of signal processing techniques based on the analysis/synthesis of a signal. Figure 2.1 is a high level model of an analysis/synthesis based speech coder.

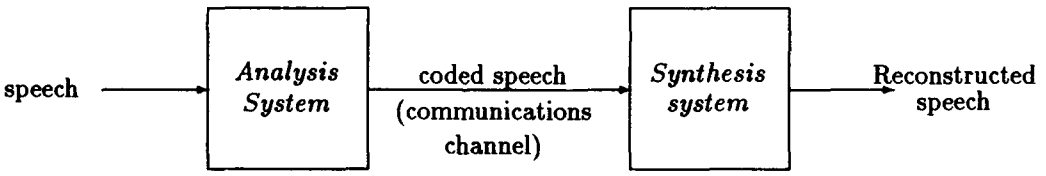


Figure 2.1. High level model of analysis-synthesis speech coder.

The analysis system of a speech coder is responsible for extracting parameters that best represent a stationary portion of the speech waveform. These parameters (which may be quantized, depending on the application) are then passed through some sort of communications channel (again, depending on the application) to the synthesis system. Here, the parameters are used to construct a speech waveform that sounds nearly identical to the original speech (13:168-169).

The Speech Signal

Figure 2.2 is a typical portion of a speech signal.

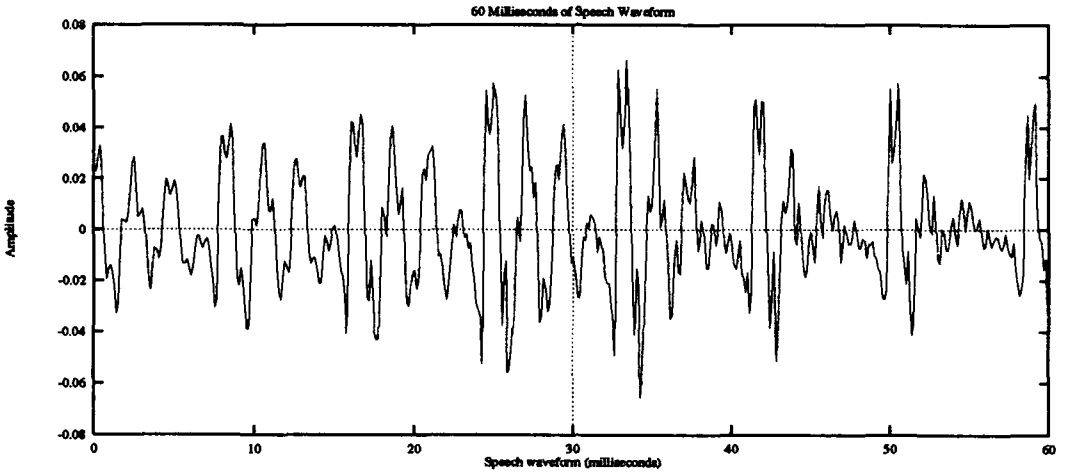


Figure 2.2. Typical portion of a digitized 8kHz speech signal

The characteristic properties of the speech waveform are such that short segments of it exhibit stationarity and, as such, the speech waveform may be represented as a sum of several sine waves of different amplitudes, frequencies, and phases. Such an equation may be written as

$$s(t) = \sum_{k=1}^{N} A_k \sin(2\pi f_k t + \Theta_k) \tag{2.1}$$

where A_k is the amplitude of the k^{th} sine wave, f_k is the frequency of the k^{th} sine wave, and Θ_k is the phase of the k^{th} sine wave (19:2).

Letting ω_k equal $2\pi f_k$ and substituting a cosine wave for the sine wave (with a new phase ϕ_k) in (2.1), a similar equation is derived:

$$\bar{s}(t) = \sum_{k=1}^{N} A_k \cos(\omega_k t + \phi_k.) \tag{2.2}$$

Equation (2.2) represents the speech as a sum of cosine waves of various amplitudes, frequencies, and phases. STC is based on this representation of the speech waveform.

For this thesis, the distinction between *voiced* and *unvoiced* speech needs to be made. For voiced speech (speech accompanied by phonation–see (16:ch3)), the excitation waveform of Figure 1.1 is a periodic pulse train where the pulses are equally spaced. The separation of the pulses determines the speaker's pitch. In general, the pulses of the excitation waveform for female speakers are closer together than that of male speakers, resulting in a higher pitch for females. Speech that is perfectly voiced can be represented by a sum of harmonic sine waves, with one sine wave per pitch period. Therefore, female speech will generally be represented by a sum of fewer sine waves than male speech, since the pitch is of higher frequency and therefore fewer harmonic pitch periods exist. For unvoiced speech, the excitation waveform of Figure 1.1 is not a periodic signal but rather a noiselike signal. The underlying sine waves will therefore be aharmonic and will not correspond to the speaker's pitch. The motivation for the representation of speech as a sum of underlying sine waves is derived from the harmonic properties of voiced speech (13:167-168).

The Sinusoidal Transform Coder

STC is an analysis-synthesis speech coder which encodes speech based upon the sinusoidal model described above. On the analysis side of STC, the underlying sine waves are located and their characteristics encoded for transmission to the synthesis side of STC, where the speech waveform is reconstructed.

Speech data is processed within STC in "frames", which are quasi-stationary portions of the speech waveform. The frame size must therefore be chosen so as not to be too large that the speech signal within the frame changes its periodic shape. STC normally operates at a 20 millisecond frame rate, which means that each frame contains a 20 millisecond portion of the input speech waveform. At a sampling rate of 8000 Hz (8000 samples per second), 20 milliseconds is equivalent to 160 samples of the digitized speech signal:

$$\frac{20}{1000} \text{ seconds} \times \frac{8000 \text{ samples}}{\text{second}} = 160 \text{ samples}.$$

The transmission rate of the data crossing the communications channel is measured in bits per second (bps). The bps rate indicates how many bits of information are sent across the communications channel per second in order to reconstruct (synthesize) the speech. Transmission rates vary from 1200-12000bps. At a transmission rate of 4800 bps, 96 bits of information are passed per each 20 millisecond frame:

$$\frac{4800 \text{ bits}}{\text{second}} \times \frac{20}{1000} \text{ seconds} = 96 \text{ bits}$$

A block diagram of STC is presented at the end of this chapter in order to provide the reader a broad overview of STC's entire operation. The block diagram is based on the operation of STC within a real-time environment, with an outer frame size of 20 milliseconds (each outer frame consists of two inner 10 millisecond frames).

Analysis System of STC

The analysis system of STC is responsible for extracting parameters from each speech frame and passing these parameters, or coded versions of them, to the synthesis portion of STC (13:168). The cepstral coefficients are computed within the analysis portion of STC. They are then converted into channel gains (13:196-197) which are passed to the synthesis system where the speech is reconstructed.

Pick peaking routine: Obtaining the Amplitudes and Frequencies of the Underlying Sine Waves. Since the solution of the cepstral coefficients derived under this thesis is based on the amplitudes and frequencies of the underlying sine waves of the speech waveform, it is appropriate to discuss how these are derived.

In order to locate the underlying sine wave amplitudes and frequencies, the magnitude spectrum (a one-dimensional array structure within STC) is searched incrementally, starting from a predetermined cutoff (such as 0 Hz) to a predetermined limit (such as 4000 Hz). At each point in

the spectrum, a check is made for a change in slope: if a point in the discrete spectrum is greater than its two nearest neighbors, or if a point is equal to one of its neighbors while greater than the other, then the frequency where that point occurs (a "peak" frequency) is taken to be a frequency of an underlying sine wave. The amplitude of the magnitude spectrum at that point is the amplitude of the corresponding sine wave.

Typically, the number of underlying sine waves for 10 milliseconds of speech lies between 15-60, and depends upon the pitch of the speaker.

The SEEVOC Technique: an Alternative Method for Locating the Underlying Sine Waves. A technique developed by D.B. Paul of MIT Lincoln Laboratories provides an alternative method of locating the underlying sine waves of a speech segment that avoids a problem of the straight-forward peak picking routine described above. The SEEVOC (Spectral Envelope Estimation Vocoder) technique disregards the low level peaks present in the magnitude of the STFT on the assumption that these peaks do not indicate the presence of an underlying sine wave but rather are a byproduct of the windowed Fourier operation, which tends to introduce low amplitude sidelobes (17:787).

The SEEVOC peak-finding technique locates peaks in the STFT magnitude by scanning the magnitude starting at the fundamental frequency. A interval around the fundamental frequency is searched for the maximum amplitude; the frequency of that amplitude is taken to be the frequency of an underlying sine wave. The search interval is then shifted by the speaker's average pitch and the procedure is repeated until the entire STFT magnitude has been searched. This method ensures that only one sine wave is located per pitch period, and that any peaks lower than the maximum peak within a search interval will not be construed as the result of an underlying sine wave (13:383) (17).

Current Computation of the Cepstral Coefficients withing STC. The cepstral coefficients are computed within STC based upon the following equation:

$$c_m = \frac{1}{\pi}\int_0^{\pi} \log A_s(\omega)\cos(m\omega)\ d\omega \qquad m = 0, 1, \ldots \tag{2.3}$$

where c_m are the cepstral coefficients and $\log A_s(\omega)$ represents the logarithm of the magnitude of the system function in figure 1.1.

Equation 2.3 is derived from Oppenheim and Schafer's definition of the complex cepstrum (see Section 1.5.1). The discrete Fourier transform (14:45) of the log of system function $H_s(\omega)$ is given by

$$\log H_s(\omega) = \sum_{m=-\infty}^{\infty} c_m e^{-jm\omega}. \tag{2.4}$$

Using Euler's identity, the above is equivalent to

$$\log H_s(\omega) = \sum_{m=-\infty}^{\infty} c_m \cos(m\omega) - j\sum_{m=-\infty}^{\infty} c_m \sin(m\omega). \tag{2.5}$$

Noting that $\log H_s(\omega) = \log\{|H_s(\omega)|e^{j\Phi(\omega)}\} = \log|H_s(\omega)| + j\Phi(\omega)$, the following is derived from equation 2.5:

$$\log|H_s(\omega)| = \sum_{m=-\infty}^{\infty} c_m \cos(m\omega). \tag{2.6}$$

Letting $A_s(\omega)$ represent $|H_s(\omega)|$ then the Fourier coefficient c_m's of equation 2.6 are given by

$$c_m = \frac{1}{\pi}\int_0^{\pi} \log A_s(\omega)\cos(m\omega)\ d\omega \tag{2.7}$$

which is the same as equation 2.3.

Within STC, the integral in equation 2.7 is approximated by the rectangular rule summation

$$c_m = \frac{\pi}{Y} \sum_{y=1}^{Y} \log A_s(\omega_y) \cos(m\omega_y) \qquad m = 0, 1, \ldots \tag{2.8}$$

where Y is the number of points in the magnitude spectrum. The rectangular rule of numerical integration applied here assumes correctly that the ω_ys (the sampled frequencies) are equally spaced.

Equation 2.8 is computed a total of M times, where M is the number of desired cepstral coefficients. Equation 2.8 is itself a summation of Y terms. Therefore, the computational complexity is of $O(MY)$. In STC, $A_s(\omega)$ is an envelope constructed from a linear interpolation of the SEEVOC peaks (13:183). The formation of the SEEVOC envelope via a linear interpolation technique is a computationally efficient algorithm of $O(Y)$ complexity. Therefore, the total computational complexity in constructing the SEEVOC envelope and then computing the cepstral coefficients based upon this envelope is $O(Y) + O(MY)$.

Parameters passed to Synthesis System

The purpose of the analysis system is to derive values for a minimum set of parameters which are passed to the synthesis system for reconstruction of the speech signal. Values for the parameters are passed to the synthesis system for every 20 milliseconds of speech. As derived above, speech coding at 4800bps of 20 milliseconds of speech allows for 96 bits of information. Within STC, these bits are normally utilized as follows:

pitch:	14 bits
voicing probability:	4 bits
spectrum envelope:	69 bits
miscellaneous:	9 bits
Total:	96 bits

The majority of the allowed bits are used to code a spectrum envelope, which is derived directly from the cepstral coefficients. The pitch and voicing probability (whether the speech is

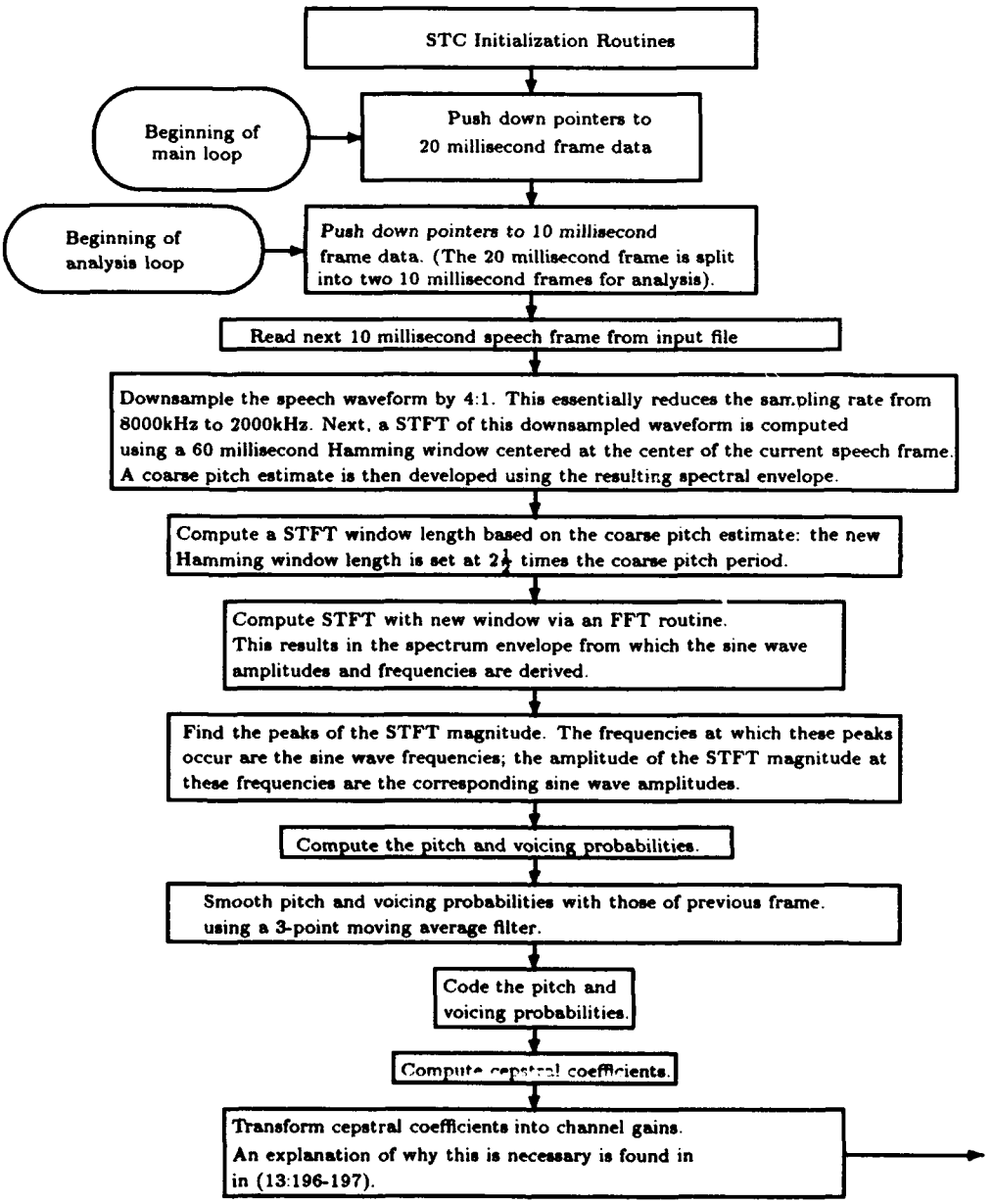
classified as voiced or unvoiced) were discussed previously in this chapter. Detailed information on the importance of pitch and voicing probability are available in (16).

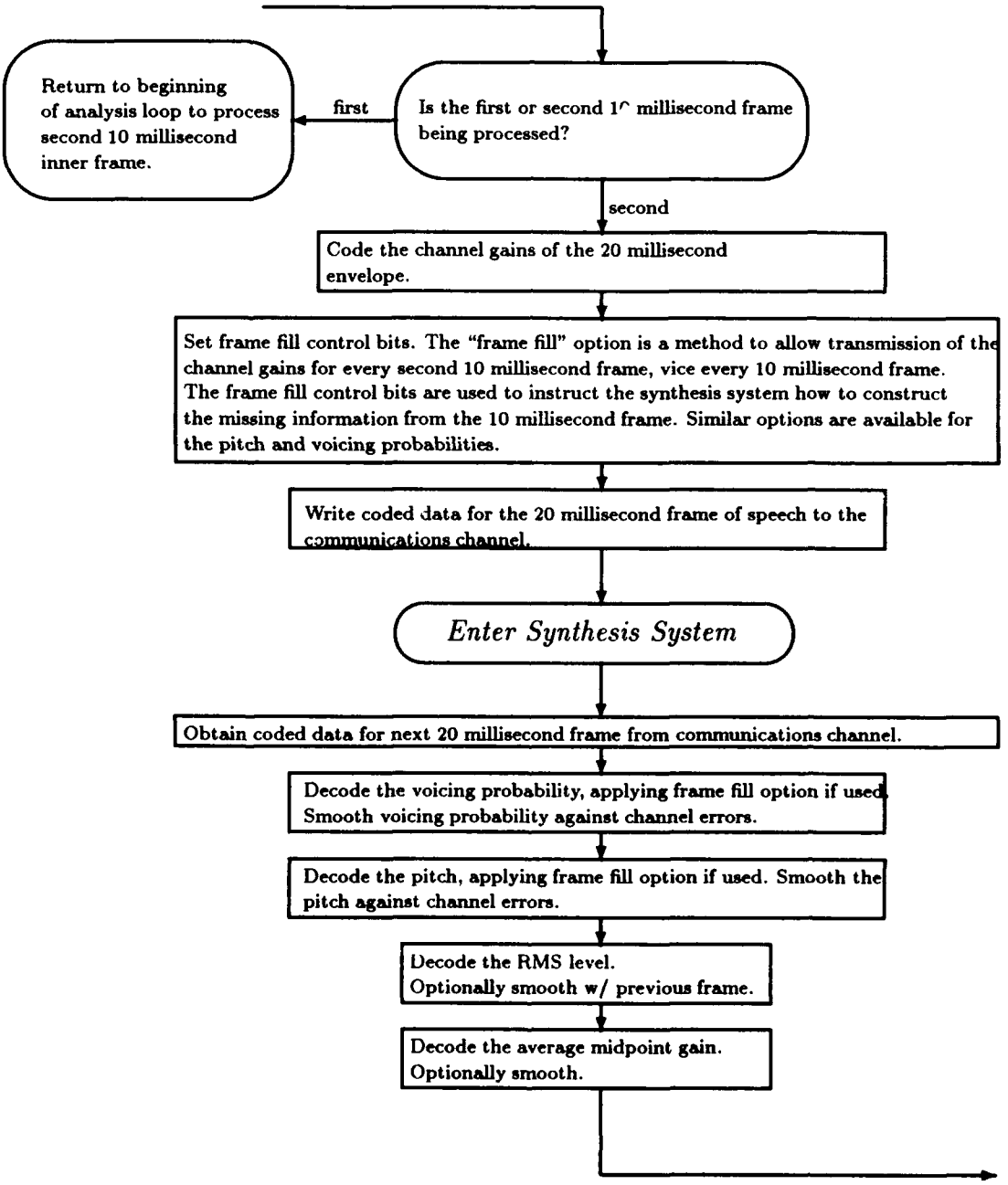
Synthesis System of STC

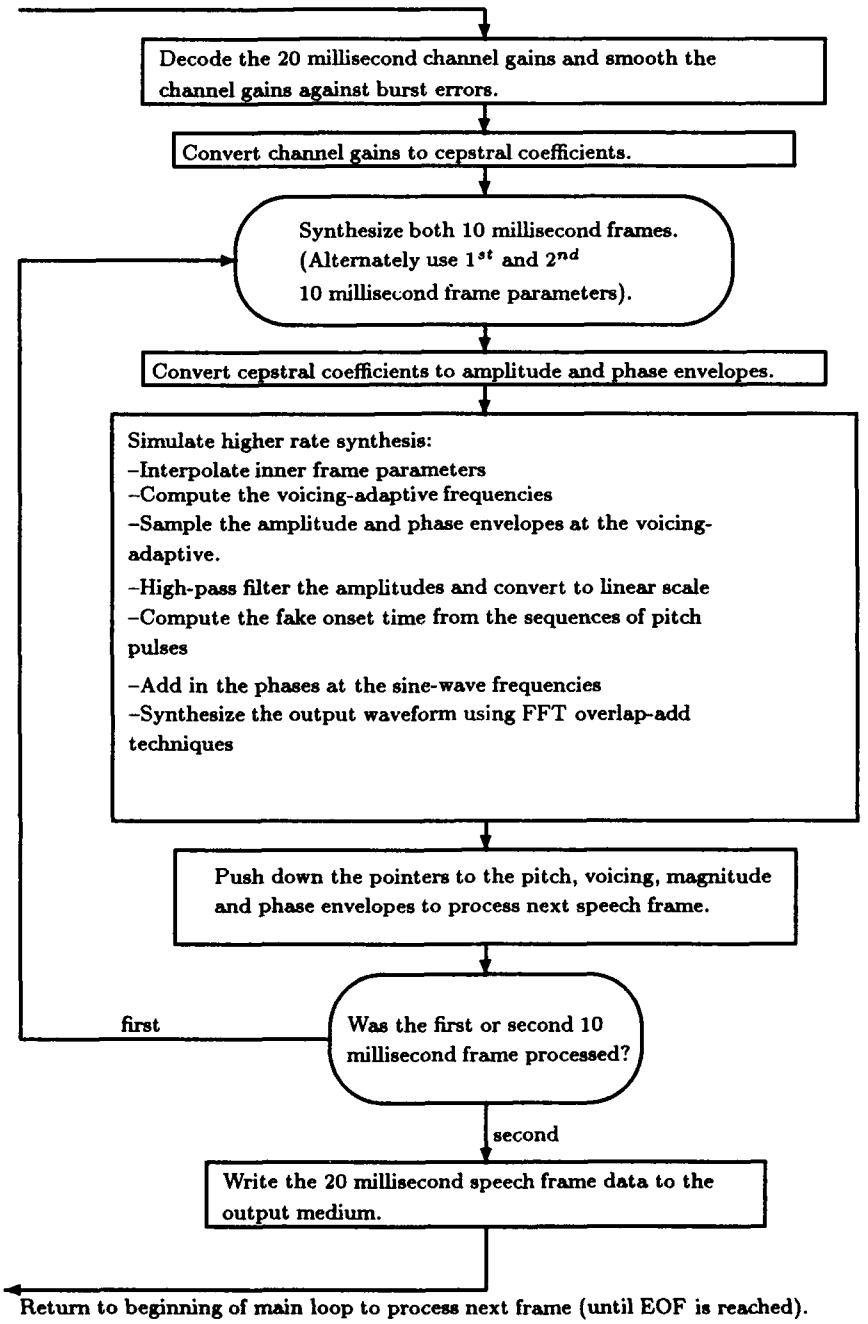
The synthesis portion of the STC constructs a waveform by generating sine waves based on the parameters passed from the analysis system. The sine waves for each frame are then summed together to obtain synthetic speech output which is perceptually equivalent to the original speech (10). The synthesis system of STC is outlined within the STC block diagram at the end of this chapter.

A good review of the operation of the synthesis system within STC is found in (13). Since this thesis is not directed toward the synthesis operation of STC, the reader is directed to (13) for more information.

Block Diagram of STC




Return to beginning of analysis loop to process second 10 millisecond inner frame.
first
Is the first or second 10 millisecond frame being processed?
second
Code the channel gains of the 20 millisecond envelope.
Set frame fill control bits. The "frame fill" option is a method to allow transmission of the channel gains for every second 10 millisecond frame, vice every 10 millisecond frame. The frame fill control bits are used to instruct the synthesis system how to construct the missing information from the 10 millisecond frame. Similar options are available for the pitch and voicing probabilities.
Write coded data for the 20 millisecond frame of speech to the communications channel.
Enter Synthesis System
Obtain coded data for next 20 millisecond frame from communications channel.
Decode the voicing probability, applying frame fill option if used. Smooth voicing probability against channel errors.
Decode the pitch, applying frame fill option if used. Smooth the pitch against channel errors.
Decode the RMS level.
Optionally smooth w/ previous frame.
Decode the average midpoint gain.
Optionally smooth.


Decode the 20 millisecond channel gains and smooth the channel gains against burst errors.
Convert channel gains to cepstral coefficients.
Synthesize both 10 millisecond frames. (Alternately use 1st and 2nd 10 millisecond frame parameters).
Convert cepstral coefficients to amplitude and phase envelopes.
Simulate higher rate synthesis:
-Interpolate inner frame parameters
-Compute the voicing-adaptive frequencies
-Sample the amplitude and phase envelopes at the voicing-adaptive.
-High-pass filter the amplitudes and convert to linear scale
-Compute the fake onset time from the sequences of pitch pulses
-Add in the phases at the sine-wave frequencies
-Synthesize the output waveform using FFT overlap-add techniques
Push down the pointers to the pitch, voicing, magnitude and phase envelopes to process next speech frame.
Was the first or second 10 millisecond frame processed?
first
second
Write the 20 millisecond speech frame data to the output medium.
Return to beginning of main loop to process next frame (until EOF is reached).

III. Methodology.

Introduction

This chapter provides the methodology for answering the first research question posed for this thesis: Can a correct algorithm be derived for a direct solution of the cepstral coefficients based on fitting a cepstral model to the measured speech data?

To answer this question, a correct cepstral model is first presented, based on the cepstral definition of Chapter 1. The background for this development is found in McAulay and Quatieri (13) and Oppenheim and Schafer (14). This cepstral model is then fit to the measured speech data, which consists of amplitudes and frequencies of the underlying cosine waves of digitized speech waveforms (equation 2.2), resulting in a mathematical solution for the cepstral coefficients.

The second research question posed for this thesis is also discussed in this chapter: What mathematical approximations to the derived algorithm are possible? Which are the fastest and most efficient, to enable execution within a real-time environment?

Several mathematical approximations to the cepstral solution are derived in this Chapter, and their relative complexities and execution times discussed. References are provided for all suggested approximations and suggested implementations within the *C* programming language.

The results of these efforts are presented in Chapter 4.

Theoretical Background

Referring to Figure 1.1, the system function, $H_s(\omega)$, of the vocal tract filters can be assumed to be minimum phase (14) and equation 1.1 defines the cepstral envelope. This equation is redisplayed below for clarity:

$$c_m = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log H_s(\omega) e^{j\omega m} d\omega. \tag{3.1}$$

The $\log H_s(\omega)$ is expanded in Fourier series as follows (14:45):

$$\log H_s(\omega) = \sum_{m=-\infty}^{+\infty} c_m e^{-jm\omega} \tag{3.2}$$

where c_m are the cepstral coefficients and $j = \sqrt{-1}$. Using Euler's identity, this equation becomes:

$$\log H_s(\omega) = \sum_{m=-\infty}^{+\infty} c_m \cos(m\omega) - j \sum_{m=-\infty}^{+\infty} c_m \sin(m\omega). \tag{3.3}$$

Finally, exploiting the even symmetric properties of the spectrum envelope, equation 3.3 evolves as

$$\log H_s(\omega) = c_0 + 2 \sum_{m=1}^{\infty} c_m \cos(m\omega) - 2j \sum_{m=1}^{\infty} c_m \sin(m\omega). \tag{3.4}$$

Noting again that $\log H_s(\omega) = \log\{|H_s(\omega)|e^{j\Phi(\omega)}\} = \log[|H_s(\omega)|] + j\Phi(\omega)$ (14), where $\Phi(\omega)$ is the phase of the system function, it is observed from the above that the log magnitude of the system function, also known as the cepstral amplitude envelope (13:181), is

$$\log |H_s(\omega)| = c_0 + 2 \sum_{m=1}^{\infty} c_m \cos(m\omega), \tag{3.5}$$

while the phase envelope is

$$\Phi(\omega) = -2 \sum_{m=1}^{\infty} c_m \sin(m\omega). \tag{3.6}$$

To obtain the cepstral coefficients, the difference between the measured values of the log magnitude of the speech data being processed by STC and the cepstral amplitude envelope (equation 3.5) must be minimized. Letting $\log A(\omega)$ represent the log magnitude of the spectrum envelope for the actual speech data, the problem is to fit the cepstral amplitude envelope (3.5) to $\log A(\omega)$. This can be accomplished through use of the mean-squared error criterion, which is the average value of the squared error between two signals. The mean-squared error is appropriate as it is easy to work with analytically, plus it accentuates large differences while minimizing small differences. This is

comparable to how the human ear functions: the ear notices large errors in a speech signal but tends to mask or ignore small errors.

For the problem in question, the mean-squared error may be written as:

$$f = \frac{1}{2\pi}\int_0^{2\pi}\left[\log A(\omega) - \log|H_s(\omega)|\right]^2 d\omega \tag{3.7}$$

where $A(\omega)$ is the actual measured amplitude of the spectral envelope of the speech data at frequency ω and $\log|H_s(\omega)|$ is equal to equation 3.5. Substituting equation 3.5 into the above yields

$$f = \frac{1}{2\pi}\int_0^{2\pi}\left[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))\right]^2 d\omega. \tag{3.8}$$

To minimize the mean-squared error, a basic law of calculus is employed, which is that a function of one variable has an extrema (maxima or minima) where its derivative evaluates to zero. Since equation 3.8 is a function of an infinite number of variables (the variables being the cepstral coefficients, $c_0, c_1, ..., c_\infty$), the partial derivatives (derivatives with respect to $c_l, l = 0, 1, ...\infty$) are used to determine the extrema of equation 3.8. The partial derivative of equation 3.8 is derived as follows:

$$\begin{aligned}
\frac{\partial f}{\partial c_0} &= \frac{\partial}{\partial c_0}\left\{\frac{1}{2\pi}\int_0^{2\pi}\left[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))\right]^2 d\omega\right\} \\
&= \frac{1}{2\pi}\int_0^{2\pi}\frac{\partial}{\partial c_0}[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))]^2 d\omega \\
&= \frac{2}{2\pi}\int_0^{2\pi}\left[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))\right]\frac{\partial}{\partial c_0}\left[\log A(\omega) - c_0 - 2\sum_{m=1}^{\infty} c_m \cos(m\omega)\right] d\omega \\
&= \frac{1}{\pi}\int_0^{2\pi}\left[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))\right](-1) \quad d\omega \\
&= -\frac{1}{\pi}\int_0^{2\pi}\log A(\omega)\, d\omega + \frac{1}{\pi}\int_0^{2\pi} c_0\, d\omega + \frac{2}{\pi}\int_0^{2\pi}\sum_{m=1}^{\infty} c_m \cos(m\omega)\, d\omega \\
&= -\frac{1}{\pi}\int_0^{2\pi}\log A(\omega)\, d\omega + \frac{1}{\pi}c_0\left[\omega\right]_0^{2\pi} + \frac{2}{\pi}\left[\sum_{m=1}^{\infty}\frac{c_m}{m}\sin(m\omega)\right]_0^{2\pi} \\
&= -\frac{1}{\pi}\int_0^{2\pi}\log A(\omega)\, d\omega + 2c_0 + 0.
\end{aligned}$$

Setting the resulting equation equal to zero yields the following for c_0:

$$c_0 = \frac{1}{2\pi}\int_0^{2\pi} logA(\omega)d\omega. \tag{3.9}$$

A similar process is followed in taking the derivative of equation 3.8 with respect to c_l, $l \neq 0$:

$$\begin{aligned}
\frac{\partial f}{\partial c_l} &= \frac{\partial}{\partial c_l}\left\{\frac{1}{2\pi}\int_0^{2\pi}\left[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))\right]^2 d\omega\right\} \\
&= \frac{1}{2\pi}\int_0^{2\pi}\frac{\partial}{\partial c_l}\left[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))\right]^2 d\omega \\
&= \frac{2}{2\pi}\int_0^{2\pi}\left[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))\right]\frac{\partial}{\partial c_l}\left[\log A(\omega) - c_0 - 2\sum_{m=1}^{\infty} c_m \cos(m\omega)\right] d\omega \\
&= \frac{1}{\pi}\int_0^{2\pi}\left[\log A(\omega) - (c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega))\right]\left[-2\cos(l\omega)\right] d\omega \\
&= -\frac{2}{\pi}\int_0^{2\pi}\log A(\omega)\cos(l\omega)d\omega + \frac{2}{\pi}\int_0^{2\pi} c_0\cos(l\omega)d\omega + \frac{4}{\pi}\int_0^{2\pi}\cos(l\omega)\sum_{m=1}^{\infty} c_m\cos(m\omega)d\omega \\
&= -\frac{2}{\pi}\int_0^{2\pi}\log A(\omega)\cos(l\omega)d\omega + \frac{2c_0}{l\pi}\sin(l\omega)d\omega\Big]_0^{2\pi} + \frac{4}{\pi}c_l\int_0^{2\pi}\cos^2(l\omega)d\omega \\
&= -\frac{2}{\pi}\int_0^{2\pi}\log A(\omega)\cos(l\omega)d\omega + 0 + \frac{4}{\pi}c_l\left[\frac{\omega}{2} + \frac{\sin(2l\omega)}{4l}\right]_0^{2\pi} \\
&= -\frac{2}{\pi}\int_0^{2\pi}\log A(\omega)\cos(l\omega)d\omega + 4c_l.
\end{aligned}$$

Again, setting the resulting equation equal to zero yields the following for $c_l, l \neq 0$:

$$c_l = \frac{1}{2\pi}\int_0^{2\pi} logA(\omega)\cos(l\omega)d\omega \qquad l = 0, 1, \ldots \tag{3.10}$$

Since $\cos(0) = 1$, equations 3.9 and 3.10 are equivalent for $k = 0$. Therefore, equation 3.10 may be used to represent the cepstral coefficients, c_l, $l = 0...\infty$. Note that when the first derivative of a function is equal to zero, this indicates the presence of either a maxima or a minima.

Minima vs. Maxima

In order to ensure minima values are located, the second partial derivatives of equation 3.7 must be evaluated. This is a more involved task for multivariable equations than for one-variable equations. In the case under consideration, there are an infinite number of partial derivatives to consider. The procedures found in (3) may be used to determine whether the cepstral coefficients result in a local maxima, local minima, or a point of inflection. The base theorem for this problem is found in (3:194) and states the following regarding the case under consideration:

Theorem 1. *Let U be an open set in $\mathbf{R^n}$ and f: $U \rightarrow \mathbf{R}$ a function having continuous second order partial derivatives. Let c_l be a critical point of f and let $HM(f)(c_l)$ be the Hessian matrix of f and c_l.*

1. *If $HM(f)(c_l)$ is positive definite, then c_l is a local minimum.*
2. *If $HM(f)(c_l)$ is negative definite, then c_l is a local maximum.*
3. *If $HM(f)(c_l)$ is indefinite, then c_l is neither a local maximum nor a local minimum.*

In the above theorem, f refers to equation 3.7 and c_l, $(l = 0, 1....)$ refers to the cepstral coefficients. Per the above theorem, the cepstral coefficients will result in local minima *if* the Hessian matrix is positive definite. The Hessian matrix may be formed by following the procedures found in (3:138-140). The Hessian matrix is built from the gradient of f at c_l, which is a vector consisting of the first partial derivatives of f with respect to c_l, $l = 0, 1...$:

$$
\begin{aligned}
(\nabla f)(c_l) &= \left(\frac{\partial f}{\partial c_0}, \frac{\partial f}{\partial c_1}, \frac{\partial f}{\partial c_2}, \cdots, \right) \\
&= \left(-\frac{1}{\pi}\int_0^{2\pi} \log A(\omega)\, d\omega + 2c_0, -\frac{2}{\pi}\int_0^{2\pi} \log A(\omega)\cos(\omega)\, d\omega + 4c_1, -\frac{2}{\pi}\int_0^{2\pi} \log A(\omega)\cos(\omega)\, d\omega + 4c_2, \cdots \right).
\end{aligned}
$$

The second order partial derivative of $\frac{\partial f}{\partial c_i}$, $(i = 0, 1, ...)$, is defined as $\frac{\partial}{\partial c_j}(\frac{\partial f}{\partial c_i})$, $(i, j = 0, 1, ...)$ and is commonly denoted by $\frac{\partial^2 f}{\partial c_j \partial c_i}$. In order to find $HM(f)$, the partial derivatives of (∇f) are taken with respect to c_j, $(j = 0, 1, ...)$. This process results in the following:

$$
HM(f) = \begin{bmatrix}
\frac{\partial^2 f}{\partial c_0 \partial c_0} & \frac{\partial^2 f}{\partial c_1 \partial c_0} & \frac{\partial^2 f}{\partial c_2 \partial c_0} & \cdots \\
\frac{\partial^2 f}{\partial c_0 \partial c_1} & \frac{\partial^2 f}{\partial c_1 \partial c_1} & \frac{\partial^2 f}{\partial c_2 \partial c_1} & \cdots \\
\frac{\partial^2 f}{\partial c_0 \partial c_2} & \frac{\partial^2 f}{\partial c_1 \partial c_2} & \frac{\partial^2 f}{\partial c_2 \partial c_2} & \cdots \\
\vdots & \vdots & \ddots & \vdots
\end{bmatrix}.
$$

Taking the second partial derivatives of (∇f) (equation 3.11) and putting them into matrix form as above yields the following matrix for $HM(f)$:

$$
\begin{bmatrix}
2 & 0 & 0 & \cdots & 0 \\
0 & 4 & 0 & \cdots & 0 \\
0 & 0 & 4 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 4
\end{bmatrix}.
$$

According to Theorem 1 above, if $HM(f)$ is positive definite, then setting the first partial derivatives of equation 3.7 equal to zero result in a local minima, as desired. To determine if $HM(f)$ is positive definite, the following theorem, found in (21:22), is employed:

Theorem 2. *HM is positive definite if and only if there exists a lower-triangular matrix G with positive main-diagonal entries, such that $HM = GG^T$.*

A diagonal matrix, such as the above, is trivial to factor into the form GG^T. For the above, G is as follows:

$$\begin{bmatrix} \sqrt{2} & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{bmatrix}$$

Since, by theorem 2, HM is positive definite, then by theorem 1, the cepstral coefficients as computed do result in the desired minimum values for the mean-squared error in equation 3.7.

Application to Speech Coding

Equation 3.10 represents the cepstral coefficients for a continuous spectral envelope; in speech coding, however, the spectral envelope is not continuous, as the speech waveform from which it is derived (through a Fourier transform operation) is digitized. Therefore, equations for the cepstral coefficients must be derived based on a discrete spectral envelope. This is done by discretizing equation 3.8 as

$$\bar{f} = \sum_{k=1}^{K} \left[log A(\omega_k) - c_0 - 2 \sum_{m=1}^{M-1} c_m \cos(m\omega_k) \right]^2. \tag{3.11}$$

where K is the number of peaks in the spectral envelope of the measured speech data, and M is the number of cepstral coefficients.

Similar to that encountered in Chapter 2, the rectangular rule of numerical integration is applied in the above equation with the assumption that the ω_ks are equally spaced. This will be the case if the peaks of the STFT magnitude are equally spaced, which only occurs when the speech is perfectly voiced so the underlying sine waves are harmonic. Thus, development under this

thesis is based on the properties of perfectly voiced speech; whether the development is adequate for non-perfectly voiced speech is discussed in the next chapter.

In equation 3.11, the number of peaks of the spectral envelope may be obtained within STC using the procedures described in Chapter 2, either via the pick peaking technique or the SEEVOC peak finding technique. Either way, each peak location is taken to define the frequency of an underlying sine wave of the speech waveform and the amplitude of the spectral envelope at that frequency is the amplitude of the underlying sine wave. Within STC, a maximum value for K is established (such as $K = 100$). The cepstral length, M, is a design parameter which is varied depending on the coder rate. (13:193). For a data rate of 4800bps, the cepstral length is typically truncated at 28 cepstral values ($M = 28$). The length of the cepstral envelope may be truncated at this relatively low number due to the nature of the cepstral transformation. As mentioned in Chapter 1, the cepstral envelope results from taking an inverse Fourier transform of the logarithm of the system function. Referring back to Figure 1.1. $H_s(\omega)$ is equal to $H_g(\omega)H_v(\omega)$. Noting again that $\log H_s(\omega) = \log[|H_s(\omega)|] + j\Phi(\omega)$, this results in

$$\begin{aligned} \log H_s(\omega) &= \log H_g(\omega)H_v(\omega) \\ &= \log H_g(\omega) + \log H_v(\omega) \\ &= \log|H_g(\omega)| + \log|H_v(\omega)| + j\Phi_g(\omega) + j\Phi_v(\omega). \end{aligned}$$

Hence,

$$\begin{aligned} \Phi_s(\omega) &= \Phi_g(\omega) + \Phi_v(\omega), \textit{and} \\ \log|H_s(\omega)| &= \log|H_g(\omega)| + \log|H_v(\omega)|. \end{aligned}$$

In the above equation $\log|H_v(\omega)|$ is a slow varying spectral envelope and $\log|H_g(\omega)|$ represents the rapidly-varying pitch-harmonic peaks (16:203). Since the spectral envelope is a slow varying component, when the further Fourier transform operation is completed, the resulting cepstral envelope is contained within the low frequency region. Since the cepstral envelope is contained

in this region, the cepstral values may be truncated at a low point within the cepstral envelope without losing significant information.

The current operation of STC holds the number of cepstral coefficients (denoted as M) fixed for the entire speech waveform. However, this approach of using a constant value of M will not work for the algorithm being developed under this thesis. Here, a cepstral model is being fit to the measured speech data, which consists of amplitudes and frequencies of underlying sine waves. It is possible that during some speech frames, when M is held constant, for there to be fewer underlying sine waves than there are cepstral coefficients. When this occurs, there are more cepstral coefficients than measurements, which mathematically permits an infinity of solutions and which in practice usually results in unstable solutions (7).

Therefore, a method needs to be determined to calculate a correct, or at least acceptable, number of cepstral coefficients for each speech frame. Referring back to Chapter 2, the amplitudes and frequencies of the underlying sine waves may be obtained by either the pick peaking routine or the SEEVOC peak finding technique. Either way, it seems possible to use the number of peaks in the STFT magnitude to determine the number of underlying sine waves, and to use this number to compute an appropriate number of cepstral coefficients. However, with the peak picking routine, some of the low-level peaks may not be the result of an underlying sine wave but rather be due to sidelobe leakage (as mentioned previously).

Also discussed in Chapter 2 was the fact that the number of underlying sine waves is based on the spacing of the pulses of the glottal excitation waveform. During voiced speech, the spacing is perfectly periodic and the number of harmonic sine waves can be determined by dividing the pitch by the length of the STFT magnitude. This gives a good estimate of the number of harmonic sine waves and should be equivalent, or close to, the number of SEEVOC peaks. The number of

cepstral coefficients could then be set to a percentage of the number of harmonics, ie.,

$$M = \alpha \times \#harmonics$$

As the pitch of the speaker increases, the number of harmonic sine waves decreases. Therefore, fewer cepstral coefficients are required to code higher-pitch speech.

Acceptable values for α can be found by comparing the original speech waveform with the output speech waveform, by listening to both waveforms and by comparing various metrics of the two waveforms. These findings are discussed in Chapter 4.

Returning to equation 3.11, note that this equation does not have a normalization constant present. Since, for the discrete case, the spectral length is controlled (K points), the normalization constant may be disgarded without any loss of information. Here, instead of the mean-squared error, equation 3.11 is more appropriately referred to as a sum-squared error.

As was done when dealing with the continuous spectral envelope, to find the cepstral coefficients that minimize equation 3.11 (discrete envelope), it is necessary to take the derivatives of equation 3.11 with respect to c_l, $(l = 0, .., M-1)$, and set these derivatives equal to 0. Taking the derivative of equation 3.11 with respect to c_l begins as follows:

$$\begin{aligned}\frac{\partial \bar{f}}{\partial c_l} &= \frac{\partial}{\partial c_l}\sum_{k=1}^{K}\Big[\log A(\omega_k) - (c_0 + 2\sum_{m=1}^{M-1} c_m \cos(m\omega_k))\Big]^2 \\ &= \sum_{k=1}^{K} 2\Big[\log A(\omega_k) - (c_0 + 2\sum_{m=1}^{M-1} c_m \cos(m\omega_k))\Big]\frac{\partial}{\partial c_l}\Big[\log A(\omega_k) - (c_0 + 2\sum_{m=1}^{M-1} c_m \cos(m\omega_k))\Big].\end{aligned}$$

For the case $l = 0$, the derivative results in

$$\frac{\partial \bar{f}}{\partial c_0} = \sum_{k=1}^{K} 2\Big[\log A(\omega_k) - c_0 - 2\sum_{m=1}^{M-1} c_m \cos(m\omega_k)\Big](-1). \tag{3.12}$$

Setting the above equation equal to 0 yields the following:

$$Kc_0 = \sum_{k=1}^{K} \log A(\omega_k) - 2\sum_{k=1}^{K}\sum_{m=1}^{M-1} c_m cos(m\omega_k). \tag{3.13}$$

For the cases of $l = 1..M-1$, the derivative of equation 3.11 becomes

$$\frac{\partial \bar{f}}{\partial c_l} = \sum_{k=1}^{K} 2\Big[\log A(\omega_k) - c_0 - 2\sum_{m=1}^{M-1} c_m \cos(m\omega_k)\Big]\Big[-2\cos(l\omega_k)\Big]. \tag{3.14}$$

Setting equation 3.14 equal to 0 and dividing both sides by -4 results in

$$\begin{aligned}
&\sum_{k=1}^{K}\Big[\log A(\omega_k) - c_0 - 2\sum_{m=1}^{M-1} c_m \cos(m\omega_k)\Big]\Big[\cos(l\omega_k)\Big] \\
= &\sum_{k=1}^{K}\Big[\log A(\omega_k)\cos(l\omega_k)\Big] - \sum_{k=1}^{K} c_0\Big[\cos(l\omega_k)\Big] - 2\sum_{k=1}^{K}\sum_{m=1}^{M-1} c_m \cos(m\omega_k)\cos(l\omega_k).
\end{aligned}$$

The third term of the above equation is nearly equal to 0 except for the case m = l, when the term reduces to $2\sum_{k=1}^{K} c_l \cos^2(\omega_k)$.

For $l = 0..M-1$, the resulting equations may be compactly summarized as follows:

$$\sum_{m=0}^{M-1}\{\sum_{k=1}^{K} \rho_m \cos(m\omega_k)\cos(l\omega_k)\}c_m = \sum_{k=1}^{K} \log A(\omega_k)\cos(l\omega_k); \quad l = 0, ..., M-1 \tag{3.15}$$

$$\text{where } \rho_m = \begin{cases} 1, & \text{m=0} \\ 2, & \text{otherwise.} \end{cases}$$

Solving $Bc = \gamma$

Equation 3.15 represents a set of M equations in M unknowns (the unknowns being the cepstral coefficients, $c_0, ..., c_{M-1}$) and are solvable as a simultaneous set of linear equations. These

equations are put into matrix form to solve the system $Bc = \gamma$, where B is a $M \times M$ matrix, γ is a $M \times 1$ vector, and c is the $M \times 1$ solution vector:

$$\begin{bmatrix} K & 2\sum\cos(\omega_k) & 2\sum\cos(2\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k) \\ \sum\cos(\omega_k) & 2\sum\cos^2(\omega_k) & 2\sum\cos(2\omega_k)\cos(\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos(\omega_k) \\ \sum\cos(2\omega_k) & 2\sum\cos(\omega_k)\cos(2\omega_k) & 2\sum\cos^2(2\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos(2\omega_k) \\ \sum\cos(3\omega_k) & 2\sum\cos(\omega_k)\cos(3\omega_k) & 2\sum\cos(2\omega_k)\cos(3\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos(3\omega_k) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum\cos((M-3)\omega_k) & 2\sum\cos(\omega_k)\cos((M-3)\omega_k) & 2\sum\cos(2\omega_k)\cos((M-3)\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos((M-3)\omega_k) \\ \sum\cos((M-2)\omega_k) & 2\sum\cos(\omega_k)\cos((M-2)\omega_k) & 2\sum\cos(2\omega_k)\cos((M-2)\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos((M-2)\omega_k) \\ \sum\cos((M-1)\omega_k) & 2\sum\cos(\omega_k)\cos((M-1)\omega_k) & 2\sum\cos(2\omega_k)\cos((M-1)\omega_k) & \cdots & 2\sum\cos^2((M-1)\omega_k) \end{bmatrix}$$

$$\times \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{M-1} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{K} \log A(\omega_k) \\ \sum_{k=1}^{K} \log A(\omega_k)\cos(\omega_k) \\ \sum_{k=1}^{K} \log A(\omega_k)\cos(2\omega_k) \\ \vdots \\ \sum_{k=1}^{K} \log A(\omega_k)\cos\big((M-1)\omega_k\big) \end{bmatrix}$$

(Note: all summations in the above matrix are $\sum_{k=1}^{K}$).

After using an appropriate algorithm (such as the Gauss-Jordan routine found in (18:32-37)) to solve this linear set of equations, the solution vector contains the cepstral coefficients which are used to code the given speech data. When these cepstral coefficients are obtained within STC, the synthesized speech is found to be nearly identical to the original speech.

Approximations for Real-time Environment

Although good computer solutions for solving the above general matrix equation do exist (the best known being Gauss-Jordan elimination), the solutions are too computationally intensive to implement within a real-time environment. The complexity of Gauss-Jordan elimination is on the order of N^3 operations (the innermost loops of the algorithm are executed N^3 times for a matrix of size $N \times N$). LU Decomposition is somewhat faster, with the computational complexity on the

order of $\frac{1}{3}N^3$ operations, but this is still too complex for implementation within a real-time system. Therefore, approximations to the above matrix must be found which allow solutions with a lower order of computational complexity (on the order of N^2 or N operations), while still maintaining a good quality of reconstructed speech.

Approximations to the B matrix are better considered by employing the following two cosine identities:

$$\begin{aligned} \cos^2(\alpha) &= \frac{1}{2}\Big(1 + \cos(2\alpha)\Big) \\ \cos(\alpha)\cos(\beta) &= \frac{1}{2}\Big(\cos(\alpha - \beta)\Big) + \Big(\cos(\alpha + \beta)\Big) \end{aligned}$$

Using these identities, and dividing both sides of the matrix equation by K, the matrix equation $Bc = \gamma$ may be rewritten as follows:

$$\frac{1}{K}\begin{bmatrix}
K & 2\sum\cos(\omega_k) & 2\sum\cos(2\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k) \\
\sum\cos(\omega_k) & 2\sum\tfrac{1}{2}\Big(1+cos(2\omega_k)\Big) & 2\sum\tfrac{1}{2}\cos(3\omega_k)+\tfrac{1}{2}\cos(\omega_k) & \cdots & 2\sum\tfrac{1}{2}\cos(M\omega_k)+\tfrac{1}{2}\cos((M-2)\omega_k) \\
\sum\cos(2\omega_k) & 2\sum\tfrac{1}{2}\cos(3\omega_k)+\tfrac{1}{2}\cos(\omega_k) & 2\sum\tfrac{1}{2}\Big(1+\cos(4\omega_k)\Big) & \cdots & 2\sum\tfrac{1}{2}\cos((M+1)\omega_k)+\tfrac{1}{2}\cos(M-3\omega_k) \\
\sum\cos(3\omega_k) & 2\sum\tfrac{1}{2}\cos(4\omega_k)+\tfrac{1}{2}\cos(2\omega_k) & 2\sum\tfrac{1}{2}\cos(5\omega_k)+\tfrac{1}{2}\cos(\omega_k) & \cdots & 2\sum\tfrac{1}{2}\cos((M+2)\omega_k)+\tfrac{1}{2}\cos(M-4\omega_k) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
\sum\cos((M-2)\omega_k) & 2\sum\tfrac{1}{2}\cos((M-1)\omega_k)+\tfrac{1}{2}\cos((M-3)\omega_k) & 2\sum\tfrac{1}{2}\cos(M\omega_k)+\tfrac{1}{2}\cos((M-4)\omega_k) & \cdots & 2\sum\tfrac{1}{2}\cos((2M-3)\omega_k)+\tfrac{1}{2}\cos(\omega_k) \\
\sum\cos((M-1)\omega_k) & 2\sum\tfrac{1}{2}\cos(M\omega_k)+\tfrac{1}{2}\cos((M-2)\omega_k) & 2\sum\tfrac{1}{2}\cos((M+1)\omega_k)+\tfrac{1}{2}\cos((M-3)\omega_k) & \cdots & 2\sum\tfrac{1}{2}\Big(1+\cos(2(M-1)\omega_k)\Big)
\end{bmatrix}$$

$$\times\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{M-1} \end{bmatrix} = \frac{1}{K}\begin{bmatrix} \sum_{k=1}^{K}\log A(\omega_k) \\ \sum_{k=1}^{K}\log A(\omega_k)\cos(\omega_k) \\ \sum_{k=1}^{K}\log A(\omega_k)\cos(2\omega_k) \\ \vdots \\ \sum_{k=1}^{K}\log A(\omega_k)\cos\big((M-1)\omega_k\big) \end{bmatrix}$$

(Note: all summations in the matrix equation are $\sum_{k=1}^{K}$).

Tridiagonal Matrix Approximation

By definition, a tridiagonal matrix is one whose elements are zero except on the diagonal of the matrix plus or minus one column. The B matrix can be simplified to tridiagonal form by analyzing the individual elements of the matrix. On the interval 0 to 2π, the cosine function has the following form (figure 3.1):

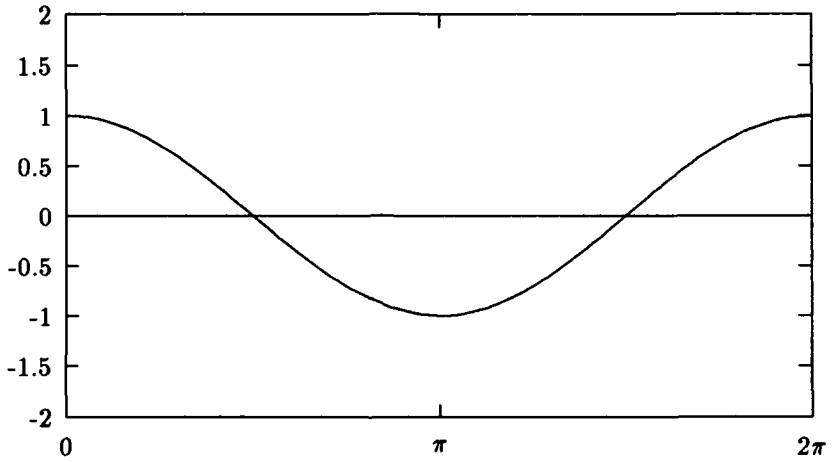


Figure 3.1. Cosine function

From figure 3.1 it is observed that, in general, when the cosine wave is evenly sampled on the interval 0 to 2π, and the samples are added together, the sum should be approximately 0 (the positive and negative values in the summation cancel each other). Noting that equation 3.11 is a numerical approximation to equation 3.7, the fact that the summation should be approximately zero is shown by evaluating the exact integral upon which it is based. For example, consider the following B matrix element:

$$2\sum_{k=1}^{K}\left[\frac{1}{2}\cos(3\omega_k)+\frac{1}{2}\cos(\omega_k)\right].$$

This element is an approximation of the following integral:

$$2\int_0^{2\pi}\left[\frac{1}{2}\cos(3\omega)+\frac{1}{2}\cos(\omega)\right]\,d\omega$$

which is evaluated as follows:

$$\begin{aligned}
& 2\int_0^{2\pi}\left[\frac{1}{2}\cos(3\omega)+\frac{1}{2}\cos(\omega)\right]\,d\omega \\
= & \int_0^{2\pi}\cos(3\omega)\,d\omega+\int_0^{2\pi}\cos(\omega)\,d\omega \\
= & \frac{1}{3}\sin(3\omega)\Big]_0^{2\pi}+\sin(\omega)\Big]_0^{2\pi} \\
= & 0.
\end{aligned}$$

Of course, the summations only evaluate to zero when the terms of the summation exactly cancel each other. But, for approximating the B matrix into a sparse matrix, such as a tridiagonal matrix, the cosine summations that occur off of the diagonal plus or minus one column may be approximated to zero.

As such, a tridiagonal approximation of the B matrix may be substituted into the equation $Bc = \gamma$ as on the following page:

$$\frac{1}{K}\begin{bmatrix}
K & 2\sum\cos(\omega_k) & 0 & \cdots & 0 & 0 & 0 \\
\sum\cos(\omega_k) & 2\sum\cos^2(\omega_k) & 2\sum\cos(2\omega_k)\cos(\omega_k) & \cdots & 0 & 0 & 0 \\
0 & 2\sum\cos(\omega_k)\cos(2\omega_k) & 2\sum\cos^2(2\omega_k) & \cdots & 0 & 0 & 0 \\
0 & 0 & 2\sum\cos(2\omega_k)\cos(3\omega_k) & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 2\sum\cos((M-3)\omega_k)\cos((M-2)\omega_k) & 2\sum\cos^2((M-2)\omega_k) & 2\sum\cos((M-1)\omega_k)\cos((M-2)\omega_k) \\
0 & 0 & 0 & \cdots & 0 & 2\sum\cos((M-2)\omega_k)\cos((M-1)\omega_k) & 2\sum\cos^2((M-1)\omega_k)
\end{bmatrix}$$

$$\times\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{M-1} \end{bmatrix} = \frac{1}{K}\begin{bmatrix} \sum_{k=1}^{K}\log A(\omega_k) \\ \sum_{k=1}^{K}\log A(\omega_k)\cos(\omega_k) \\ \sum_{k=1}^{K}\log A(\omega_k)\cos(2\omega_k) \\ \vdots \\ \sum_{k=1}^{K}\log A(\omega_k)\cos\big((M-1)\omega_k\big) \end{bmatrix}$$

Note: All summations in the above matrix are $\sum_{k=1}^{K}$.

For tridiagonal systems, solutions exist which require only $O(N)$ operations. A concise solution implemented in the C programming language using LU decomposition, forward- and back-substitution, is found in (18:47).

There is a formal technique to reduce a symmetric matrix to tridiagonal form, known as the Householder reduction. The Householder reduction method is explained in both (18:367-374) and (21:245-251). This technique was developed to find the eigenvalues and eigenvectors of a matrix, but was investigated as to its application under this thesis research.

Except for a constant factor of 2 which appears in every location of the B matrix except for those locations in column 1, the B matrix is symmetric. Taking out this constant, the Householder reduction method can be used to transform the exact matrix into tridiagonal form, a form for which the cepstral coefficients may be solved for with $O(N)$ complexity. However, the Householder algorithm is itself of $O(\frac{4}{3}N^3)$ complexity, which negates its value in reducing the computational requirements in solving for the cepstral coefficients, since the reduction method would have to be implemented for every speech frame.

Therefore, the only possible benefit of the Householder method to the problem under consideration is if it can be applied to the symbolic form of the B matrix; the resulting tridiagonal symbolic form could then be used within the coded solution of $Bc = \gamma$. There are problems with this approach. Decisions must be made within the Householder algorithm which depend on the exact values of the matrix elements (not the symbolic forms of the elements). Also, the iterative Householder transformation applied to the B matrix on the left-hand side of $Bc = \gamma$ must be applied to the vector γ as well. A technique for applying the iterative transformation to the vector γ as it is applied to the B matrix has not been developed. For these reasons, the Householder reduction method is not applicable to this problem.

Identity Matrix Approximation

Taking the concept of a tridiagonal matrix further, it is interesting to approximate the B matrix as an identity matrix. For this case, the identity matrix is denoted as I_M and by definition has all diagonal elements equal to unity and all off-diagonal elements equal to zero. The equation $Bc = \gamma$ becomes:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \times \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{M-1} \end{bmatrix} = \frac{1}{K} \begin{bmatrix} \sum_{k=1}^{K} \log A(\omega_k) \\ \sum_{k=1}^{K} \log A(\omega_k) \cos(\omega_k) \\ \sum_{k=1}^{K} \log A(\omega_k) \cos(2\omega_k) \\ \vdots \\ \sum_{k=1}^{K} \log A(\omega_k) \cos\big((M-1)\omega_k\big) \end{bmatrix}$$

In this form, $c_l = \gamma_l$ for all l. The advantage to this algorithm is that it is very fast and extremely simple. Only γ needs to be computed and no matrix operations are involved.

Toeplitz Matrix Approximation

Toeplitz matrices occur often in DSP applications, such as in spectral estimation, linear prediction, autoregressive filter design, and error control codes (1:352). As such, many Toeplitz algorithms have been developed and it is therefore appropriate to consider a Toeplitz approximation to the B matrix.

An $N \times N$ Toeplitz matrix is composed of $2N-1$ numbers emplaced as matrix elements constant along the upper-left to lower-right diagonals of the matrix:

$$\begin{bmatrix} R_0 & R_{-1} & R_{-2} & \cdots & R_{-N+2} & R_{-N+1} \\ R_1 & R_0 & R_{-1} & \cdots & R_{-N+3} & R_{-N+2} \\ R_2 & R_1 & R_0 & \cdots & R_{-N+4} & R_{-N+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ R_{N-2} & R_{N-3} & R_{N-4} & \cdots & R_0 & R_{-1} \\ R_{N-1} & R_{N-2} & R_{N-3} & \cdots & R_1 & R_0 \end{bmatrix}$$

A Toeplitz matrix is symmetric if $R_x = R_{-x}$ for all x (18:54).

The matrix equation $Bc = \gamma$ using a symmetric Toeplitz approximation of the B matrix (developed from the first column of matrix B) is as follows:

$$\frac{1}{K}\begin{bmatrix} K & \sum\cos(\omega_k) & \sum\cos(2\omega_k) & \sum\cos(3\omega_k) & \cdots & \sum\cos((M-1)\omega_k) \\ \sum\cos(\omega_k) & K & \sum\cos(\omega_k) & \sum\cos(2\omega_k) & \cdots & \sum\cos((M-2)\omega_k) \\ \sum\cos(2\omega_k) & \sum\cos(\omega_k) & K & \sum\cos(\omega_k) & \cdots & \sum\cos((M-3)\omega_k) \\ \sum\cos(3\omega_k) & \sum\cos(2\omega_k) & \sum\cos(\omega_k) & K & \cdots & \sum\cos((M-4)\omega_k) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum\cos((M-1)\omega_k) & \sum\cos((M-2)\omega_k) & \sum\cos((M-3)\omega_k) & \sum\cos((M-4)\omega_k) & \cdots & K \end{bmatrix}$$

$$\times \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{M-1} \end{bmatrix} = \frac{1}{K}\begin{bmatrix} \sum_{k=1}^{K}\log A(\omega_k) \\ \sum_{k=1}^{K}\log A(\omega_k)\cos(\omega_k) \\ \sum_{k=1}^{K}\log A(\omega_k)\cos(2\omega_k) \\ \vdots \\ \sum_{k=1}^{K}\log A(\omega_k)\cos\big((M-1)\omega_k\big) \end{bmatrix}$$

(Note: all summations in the above matrix are $\sum_{k=1}^{K}$).

A common algorithm for solving symmetric Toeplitz problems is the Levinson algorithm, an iterative algorithm based on the bordering of the matrix. A detailed review of this algorithm

is offered in (1:353). An implementation of Levenson's algorithms done in the C programming language is found in (18:58).

Although Levinson's algorithm was designed to solve matrix equations where the matrix involved is a symmetric Toeplitz matrix, a derivation of this algorithm, by G. Rybicki (18:55), can solve matrix equations for both symmetric and nonsymmetric Toeplitz matrices. This algorithm is presented in (18:54-58).

Recently, Chan and Hansen (5) implemented a lookahead Levinson algorithm for solving symmetric indefinite and general Toeplitz systesm. Whereas Levinson's algorithm is guaranteed stable for positive definite Toeplitz systems, Chan and Hansen's implementation is numerically stable for all Toeplitz systems without "many" consecutive ill-conditioned leading principle submatrices (5). This implementation is a more sophisticated solution to the Toeplitz problem that provides stability for a wider class of Toeplitz systems. Within the context of this thesis, the lookahead algorithms might provide better solutions to the Toeplitz approximation to $Bc = \gamma$ during periods when B may become indefinite, perhaps during unvoiced speech.

A nonsymmetric Toeplitz approximation to the B matrix is also developed from the first column of the B matrix, but as the first column is shifted downward to form the remaining columns, the shifted element is multiplied by 2, so the following matrix results:

$$\frac{1}{K}\begin{bmatrix} K & 2\sum\cos(\omega_k) & 2\sum\cos(2\omega_k) & 2\sum\cos(3\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k) \\ \sum\cos(\omega_k) & K & 2\sum\cos(\omega_k) & 2\sum\cos(2\omega_k) & \cdots & 2\sum\cos((M-2)\omega_k) \\ \sum\cos(2\omega_k) & \sum\cos(\omega_k) & K & 2\sum\cos(\omega_k) & \cdots & 2\sum\cos((M-3)\omega_k) \\ \sum\cos(3\omega_k) & \sum\cos(2\omega_k) & \sum\cos(\omega_k) & K & \cdots & 2\sum\cos((M-4)\omega_k) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum\cos((M-1)\omega_k) & \sum\cos((M-2)\omega_k) & \sum\cos((M-3)\omega_k) & \sum\cos((M-4)\omega_k) & \cdots & K \end{bmatrix}$$

(Note: all summations in the above matrix are $\sum_{k=1}^{K}$).

In terms of computational complexity, the Toeplitz solution, requires $O(N^2)$ operations. With Chan and Hansen's lookahead Levinson algorithm, the complexity remains the same if no ill-conditioned submatrices are encountered. If only one ill-conditioned leading principle submatrix is encountered, the complexity increases approximately 20% (5:257). The Toeplitz solutions are not as computationally efficient as the diagonal solutions. However, they are an order of magnitude more efficient than solving a general matrix equation, which requires $O(N^3)$ complexity.

Other Toeplitz approximations are possible. One such possibility is to form the Toeplitz matrix based on *all* the data contained in the B matrix, not just the data within the first column of the B matrix. The algorithm for doing this can best be understood by first reconsidering the form of a $N \times N$ symmetric Toeplitz matrix:

$$\begin{bmatrix} R_0 & R_1 & R_2 & \cdots & R_{N-2} & R_{N-1} \\ R_1 & R_0 & R_1 & \cdots & R_{N-3} & R_{N-2} \\ R_2 & R_1 & R_0 & \cdots & R_{N-4} & R_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ R_{N-2} & R_{N-3} & R_{N-4} & \cdots & R_0 & R_1 \\ R_{N-1} & R_{N-2} & R_{N-3} & \cdots & R_1 & R_0 \end{bmatrix}$$

and the form of the B matrix:

$$\begin{bmatrix} K & 2\sum\cos(\omega_k) & 2\sum\cos(2\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k) \\ \sum\cos(\omega_k) & 2\sum\cos^2(\omega_k) & 2\sum\cos(2\omega_k)\cos(\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos(\omega_k) \\ \sum\cos(2\omega_k) & 2\sum\cos(\omega_k)\cos(2\omega_k) & 2\sum\cos^2(2\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos(2\omega_k) \\ \sum\cos(3\omega_k) & 2\sum\cos(\omega_k)\cos(3\omega_k) & 2\sum\cos(2\omega_k)\cos(3\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos(3\omega_k) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum\cos((M-3)\omega_k) & 2\sum\cos(\omega_k)\cos((M-3)\omega_k) & 2\sum\cos(2\omega_k)\cos((M-3)\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos((M-3)\omega_k) \\ \sum\cos((M-2)\omega_k) & 2\sum\cos(\omega_k)\cos((M-2)\omega_k) & 2\sum\cos(2\omega_k)\cos((M-2)\omega_k) & \cdots & 2\sum\cos((M-1)\omega_k)\cos((M-2)\omega_k) \\ \sum\cos((M-1)\omega_k) & 2\sum\cos(\omega_k)\cos((M-1)\omega_k) & 2\sum\cos(2\omega_k)\cos((M-1)\omega_k) & \cdots & 2\sum\cos^2((M-1)\omega_k) \end{bmatrix}$$

(Note: all summations in the above matrix are $\sum_{k=1}^{K}$).

The Toeplitz approximation is based on letting R_x equal the average value of all the elements of the B matrix which correspond to the locations of R_x in the Toeplitz matrix ($x = 0, 1, ...N-1$). For instance, R_0 would equal the average of all the diagonal elements in the B matrix:

$$R_0 = \frac{1}{M} \sum_{x=1}^{M} B_{xx}$$

A computer algorithm for generating this type of Toeplitz matrix is not complicated: the B matrix is modified so that all the elements corresponding to R_x (M elements total) are contained in row x of the modified B matrix (say B'). Then,

$$R_x = \frac{1}{M} \sum_{z=1}^{M} B'_{xz}$$

The drawback to this approach is the overhead: First, the B matrix must be computed (which is not necessary in the tridiagonal, identity, or first-column Toeplitz approximations), itself an $O(N^2)$ algorithm. The Toeplitz matrix is then formed from a modification to the B matrix, an $O(N)$ requirement, and finally the Toeplitz system of equations is solved with $O(N^2)$ complexity. So, this Toeplitz system is more than twice as computationally complex as the Toeplitz approximations where the Toeplitz matrix is based only upon the first column of the B matrix, and is therefore not appropriate for real-time implementation.

Computing the Cepstral Coefficients Based on a Warped Spectral Envelope

The discussion so far assumes a linear frequency scale for the underlying spectral envelope. However, as mentioned in Chapter 1, a warped frequency scale holds certain advantages for speech coding.

Within STC, the warping function parameters for operation at 4800bps typically transform a 512-point discrete linear spectral envelope into a warped spectral envelope of 293 points.

Two approaches to applying the discussed methodology on a warped frequency scale are possible. One option is to warp the entire spectral envelope before obtaining the amplitudes and frequencies of the underlying sine waves; the underline sine waves are located on the warped spectral envelope instead. The primary disadvantage of this approach is that some underlying sine waves may be lost during the warping process, ie., a sample point on the linear scale which is not converted onto the warped frequency scale may represent a frequency of an underlying sine wave. However, since the ear is less sensitive to higher frequencies, the loss of a high frequency sine wave may not be that noticeable within the synthesized speech.

The second approach is to first find the underlying sine waves on the linear scale (K sine waves per speech frame), then convert each of these "peak" sine wave frequencies onto a warped scale. This is a straight-forward operation within STC and is completed by accessing the warping table called "dft_vs_mel". This is a one-dimensional array, with the number of entries corresponding to the length of the linear spectral envelope. With a 512-point spectral envelope, mel_vs_dft table has 512 entries and mel_vs_dft[x] references the location of linear frequency x on the warped (mel) scale. The advantages of this method are that all the frequencies of the underlying sine waves are maintained during the warping process, and it is faster, since the entire spectral envelope does not have to be warped before locating the sine waves. Therefore, this method of spectral warping is recommended over the previous method.

Equipment and Support

Non-AFIT support for research under this thesis was given by Dr. Robert J. McAulay of MIT Lincoln Laboratories, who provided much time and assistance with the operation of STC and the development of the discussed algorithm. In addition, support was provided by Dr. Tim Anderson of the Biocommunications Laboratory at Wright-Patterson AFB in the use of listening and recording equipment.

To assist in the Toeplitz approximation to the cepstral solution, Per Christian Hansen and Tony Chan provided FORTRAN subroutines for solving Toeplitz systems using the lookahead Levinson method.

Work under this thesis was completed using SPARCstation 2 workstations provided by Rome Laboratories (Mr Terrance Champion) and the Air Force Office of Scientific Research (AFOSR, Dr. Jon Sjogren). All programming was accomplished in the *C* programming language and compiled under the *C* complier available on the SPARCstation 2 systems. Typesetting of the thesis document was completed using the LaTeX document preparation system. Analog cassette tapes of the synthesized speech were produced using the Entropic Signal Processing System software package and cassette recording equipment available at the Biocommunications laboratory, a subdivision of Armstrong Laboratory, Wright-Patterson AFB, Ohio.

Validation of Method

The primary method of validation of the discussed methodologies is to incorporate the different algorithms within STC and utilize listening tests to compare the synthesized speech with the original speech. Another validation method is to compare plots of the cepstral envelopes of the discussed methodologies to the cepstral envelopes currently generated within STC. The results of both of these validations are presented in the next chapter. Spectrograms of the reconstructed speech also serve to validate the algorithm, and are presented in Appendix A.

Summary

This chapter reviewed the methodology for developing an algorithm to solve for the cepstral coefficients directly from the measured underlying sine wave amplitudes and frequencies of speech waveforms. A cepstral model, based on the complex cepstrum defined by Oppenheim and Schafer in (14:770) was fit to the measured speech data by minimizing the difference between the measured

speech data and the cepstral model. Through a step-by-step mathematical analysis, a compact solution was realized, as well as possible approximations to the solution which are more computationally efficient. The results of the solution and its approximations are presented in the next chapter.

IV. Findings.

Based on the theory and methodology of the previous chapter, the resulting solutions to the matrix equation $Bc = \gamma$ were implemented in the C programming language and inserted into the STC analysis system to compute the cepstral coefficients. This chapter reviews the results of these implementations.

Findings for Research Questions

In Chapter One, three research questions were proposed for this thesis project:

1. Can a correct algorithm be derived for a direct solution of the cepstral coefficients based on fitting a cepstral model to the measured speech data?
2. If so, what mathematical approximations to the algorithm may be derived? Which are the fastest and most efficient, to enable execution within a real-time environment?
3. What are the results? Does the algorithm or any of the approximations yield reconstructed speech perceptually equivalent to the original speech?

The first question is answered positively. Chapter three reviews in detail the development of an algorithm to solve for the cepstral coefficients based on fitting a cepstral model to the measured speech data. In formulating the algorithm, Oppenheim and Schafer's definition of the complex cepstrum is used to develop the cepstral model. This cepstral model is fit to the actual speech data using a sum-squared error criterion, resulting in a set of simultaneous linear equations, denoted as $Bc = \gamma$, which is solved for the cepstral coefficients.

In answer to question two, three mathematical approximations to the solution of the cepstral coefficients are found to be possible, based on analyzing the form of the B matrix. The tridiagonal and identity approximations are based on approximating the cosine summations making up the clements of the B matrix. The cosine summations are found to tend toward zero off the diagonal

and approach unity on the diagonal. The Toeplitz approximations result from forming columns 2 through M of the Toeplitz matrix from downward shifted versions of the first column of the B matrix (the first columns of the B matrix and its Toeplitz forms are identical).

A matrix equation involving an identity matrix may be solved with $O(1)$ complexity (no matrix operations are involved). A matrix equation where the matrix is in a tridiagonal form requires $O(N)$ computations to solve. A matrix equation of a Toeplitz form has $O(N^2)$ computational complexity. All of these forms are computationally cheaper than solving a general matrix equation (such as the exact solution to $Bc = \gamma$), which requires $O(N^3)$ complexity.

Quantifying Results

In order to compare the current STC cepstral coefficients and the cepstral coefficients generated via the methods under this thesis, various metrics are defined to obtain a measurement of the difference between the two sets of cepstral coefficients.

Both the existing STC cepstral coefficients and the coefficients developed under this thesis are based on the cepstral amplitude envelope equation (equation 3.5), redisplayed below for clarity:

$$\log|H_s(\omega)| = c_0 + 2\sum_{m=1}^{\infty} c_m \cos(m\omega). \tag{4.1}$$

In the above equation, c_0 represents the average level of the cepstral envelope and the c_ms ($m > 0$) determine the shape of the cepstral envelope. In formulating the metrics, let $\hat{A}_s(\omega)$ represent a cepstral amplitude envelope developed within the context of this thesis, and $A_s(\omega)$ represent the cepstral amplitude envelope currently generated within STC.

The first metric to define is an L-2 distance between the two envelopes. Expressed in dB, the L-2 norm is developed as follows:

$$
\begin{aligned}
d_2||(A_s(\omega),\hat{A}_s(\omega))||_{L_2}^2 &= \frac{1}{\pi}\int_0^\pi \left[|20(\log_{10}2)\log_2 A_s(\omega) - 20(\log_{10}2)\log_2 \hat{A}_s(\omega)|\right]^2 d\omega \\
&= \frac{1}{\pi}\{20(\log_{10}2)\}^2 \int_0^\pi \left[(c_0-\hat{c}_0) + 2\sum_{m=1}^{M-1}(c_m-\hat{c}_m)\cos(m\omega)\right]^2 d\omega \\
&= \frac{1}{\pi}\{20(\log_{10}2)\}^2 \left[\int_0^\pi (c_0-\hat{c}_0)^2\, d\omega + 4\int_0^\pi (c_0-\hat{c}_0)\sum_{m=1}^{M-1}(c_m-\hat{c}_m)\cos(m\omega)\, d\omega \right. \\
&\qquad \left. + 4\int_0^\pi \left[\sum_{m=1}^{M-1}(c_m-\hat{c}_m)\cos(m\omega)\right]^2 d\omega\right] \\
&= \{20(\log_{10}2)\}^2\left[(c_0-\hat{c}_0)^2 + 2\sum_{m=1}^{M-1}(c_m-\hat{c}_m)^2\right].
\end{aligned}
$$

The L_1 norm is simpler than the L_2 norm, in that the integrand is not raised to the power of 2, but now the absolute value signs must be taken into account:

$$
\begin{aligned}
d_1||(A_s(\omega),\hat{A}_s(\omega))||_{L_1}^2 &= \frac{1}{\pi}\int_0^\pi \left[|20(\log_{10}2)\log_2 A_s(\omega) - 20(\log_{10}2)\log_2 \hat{A}_s(\omega)|\right] d\omega \\
&= \frac{1}{\pi}\{20(\log_{10}2)\}\int_0^\pi \log_2 A_s(\omega) - \log_2 \hat{A}_s(\omega)\, d\omega
\end{aligned}
$$

case 1: $A_s(\omega) > \hat{A}_s(\omega)$

$$
\begin{aligned}
&= \frac{1}{\pi}\{20(\log_{10}2)\}\int_0^\pi (c_0-\hat{c}_0) + 2\sum_{m=1}^{M-1}(c_m-\hat{c}_m)\cos(m\omega)\, d\omega \\
&= \{20(\log_{10}2)\}\left[c_0-\hat{c}_0 + \frac{2}{\pi}\sum_{m=1}^{M-1}\frac{(c_m-\hat{c}_m)}{m}\sin(m\omega)\right]_0^\pi \\
&= \{20(\log_{10}2)\}(c_0-\hat{c}_0)
\end{aligned}
$$

case 2: $\hat{A}_s(\omega) > A_s(\omega)$

$$
\begin{aligned}
&= -\frac{1}{\pi}\{20(\log_{10}2)\}\int_0^\pi (c_0-\hat{c}_0) + 2\sum_{m=1}^{M-1}(c_m-\hat{c}_m)\cos(m\omega)\, d\omega \\
&= -\{20(\log_{10}2)\}\left[c_0-\hat{c}_0 + \frac{2}{\pi}\sum_{m=1}^{M-1}\frac{(c_m-\hat{c}_m)}{m}\sin(m\omega)\right]_0^\pi \\
&= -\{20(\log_{10}2)\}(c_0-\hat{c}_0)
\end{aligned}
$$

$$d_1||(A_s(\omega), \hat{A}_s(\omega))||_{L_1}^2 \qquad = \qquad \{20(\log_{10} 2)\}|c_0 - \hat{c}_0|.$$

The L_1 norm is useful in determining the difference between the levels of the cepstral envelope from STC and the cepstral envelope generated under this thesis.

The final norm to consider is the L_∞ norm which is defined as

$$d_\infty||(A_s(\omega), \hat{A}_s(\omega))|| = \max_m |c_m - \hat{c}_m|,$$

and is the largest difference between corresponding cepstral coefficients of the two envelopes being compared.

Values for the above differences are indicated in the plots contained in this chapter.

Computing the Number of Cepstral Coefficients

As mentioned in the previous chapter, the current operation of STC holds the number of cepstral coefficients (M) fixed for the entire speech waveform. This method does not work for cepstral coefficients obtained under this thesis, which are obtained from the measured amplitudes and frequencies of the underlying sine waves of the speech data. The number of cepstral coefficients to compute cannot exceed the number of underlying sine waves, or else a numerically unstable solution results. As an example of a numerically unstable solution, consider figure 4.1 which compares a cepstral envelope generated within STC using $M = 28$ cepstral coefficients and the envelope generated from the method discussed in this thesis, also with 28 cepstral coefficients.

Clearly, the method under this thesis is unstable in this example since the number of data points (underlying sine waves) is much less that the number of cepstral coefficients the algorithm is trying to fit to the data. Figure 4.2 illustrates a stable solution of the same cepstral fit, this time

using only 16 cepstral coefficients for the cepstral envelope generated via the algorithm developed under this thesis.

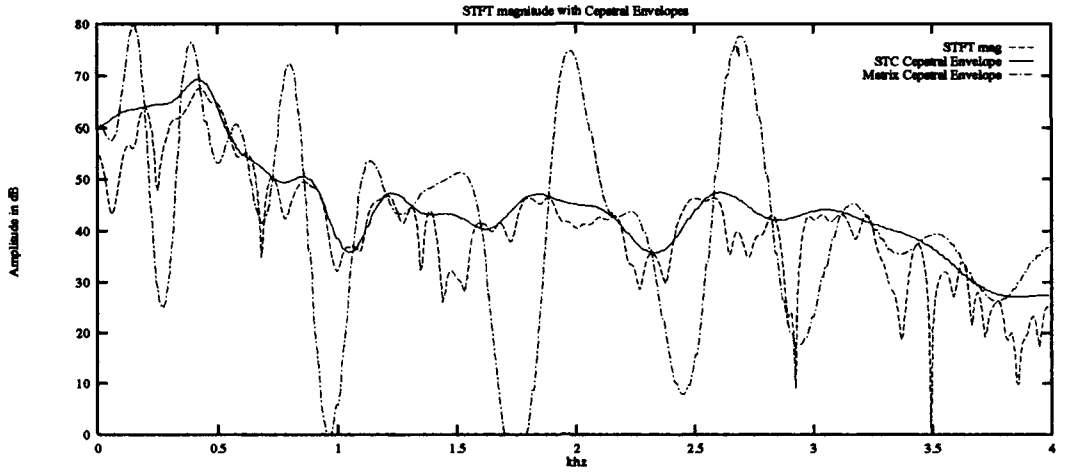


Figure 4.1. Illustration of an unstable solution of $Bc = \gamma$ due to the value of M (number of cepstral coefficients) exceeding the available number of data points (sampled sine wave frequency locations) within the speech waveform. 28 cepstral coefficients were used to construct both the STC cepstral envelope and the matrix cepstral envelope.

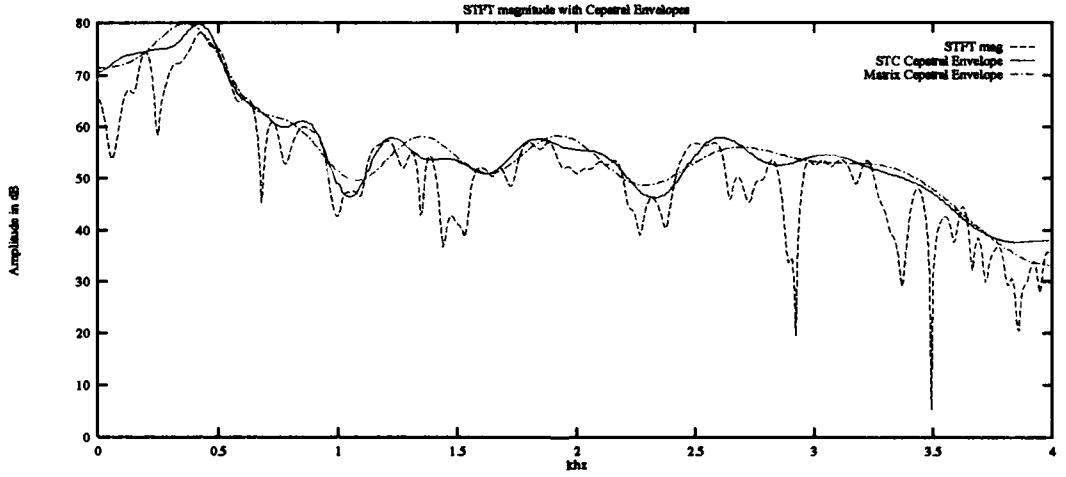


Figure 4.2. Illustration of a stable solution of the same cepstral fit as above. Here, the value of M does not exceed the available number of data points. Only 16 cepstral coefficients were used to compute the matrix cepstral envelope, vice 28 coefficients for the STC cepstral envelope.

The problem is therefore to determine a correct number of cepstral coefficients to compute based on the available speech data. The algorithm must be computationally simple, so as not to "undo" the computational benefits of the fast matrix approximations to the cepstral solutions.

Discussed in Chapter 3 is a simple equation for obtaining the number of cepstral coefficients, based on the nice properties of voiced speech, during which the underlying sine waves are harmonic and result in one underlying sine wave per pitch period. Using this property as a basis, the number of cepstral coefficients can be computed as

$$M = \alpha \times \#harmonics$$

where α is an adjustment factor to the number of harmonics. In general, as the pitch of the speaker increases, the number of harmonic sine waves decreases. Therefore, fewer cepstral coefficients are required to code higher-pitched speech.

Based on listening tests and comparing metrics of the synthesized speech using various values for α on utterances by both male and female speakers, α was found to decrease as the pitch increased. When α is set too high, "burst"-like noises within the reconstructed speech result. When α is too low, the speech is slurred.

Through listening tests and metric comparisons, a table for computing an adequate number of cepstral coefficients is developed as follows (table 4.1):

Table 4.1. Corresponding Number of Cepstral Coefficients (Order Cepstral) for Speaker Pitch.

Speaker Pitch (Hz)	Order Cepstral
$Pitch < 70$	28
$70 \leq Pitch < 80$	27
$80 \leq Pitch < 90$	26
$90 \leq Pitch < 130$	23
$130 \leq Pitch < 140$	22
$140 \leq Pitch < 150$	21
$150 \leq Pitch < 160$	20
$160 \leq Pitch < 170$	18
$170 \leq Pitch < 180$	17
$180 \leq Pitch < 200$	16
$200 \leq Pitch < 210$	15
$210 \leq Pitch < 230$	14
$230 \leq Pitch < 240$	13
$240 \leq Pitch < 270$	12
$270 \leq Pitch \leq 300$	11
$Pitch > 270$	10

Sample table for computing an acceptable order cepstral (M) for cepstral coefficients computed using the exact matrix equation $Bc = \gamma$. Tables for the Toeplitz, tridiagonal, and identity approximations differ slightly in order to maximize the reconstructed speech from the specific approximation. Also, during unvoiced speech, the number of cepstral coefficients may be set slightly slower than indicated in the table in order to prevent instablitities from occuring.

Results

As discussed in the previous two chapters, two alternatives are available to determine the amplitudes and frequencies of the underlying sine waves of speech waveforms: the straight-forward peak-picking algorithm and the SEEVOC technique. Based on metric comparisons and informal listening tests, the sine waves obtained using the SEEVOC technique yield cepstral coefficients which result in better fitting cepstral envelopes that the sine waves obtained via the peak-picking method. This is an understandable result as the peak-picking method takes into account low-level peaks within the STFT magnitude which are more than likely a function of the windowed-Fourier operation rather than an indication of an underlying sine wave (17:787). For the results presented in this section, the SEEVOC technique was used to obtain the sine wave information of the speech waveforms.

The Exact System. As expected, speech synthesized using the exact matrix represents a very close approximation to the original speech. However, the computational time is considerable. First, the B matrix must be computed (an $O(KM^2)$ requirement), then the equation $Bc = \gamma$ must be solved, which requires $O(M^3)$ complexity. The success of the exact system is highly dependent on the number of cepstral coefficients computed, especially in the case of high-pitched utterances (usually the case with female speakers). When the number of cepstral coefficients to solve for during each frame of speech is computed using the method described in the previous section, the reconstructed speech is nearly identical to the original speech. However, if the number of cepstral coefficients is set constant through the analysis-synthesis of the entire utterance, the reconstructed speech frequently contains "burst-like" noises.

The following set of plots compare cepstral envelopes generated via this method and cepstral envelopes generated currently within STC. The pitch and voicing probabilities are noted within the plots.

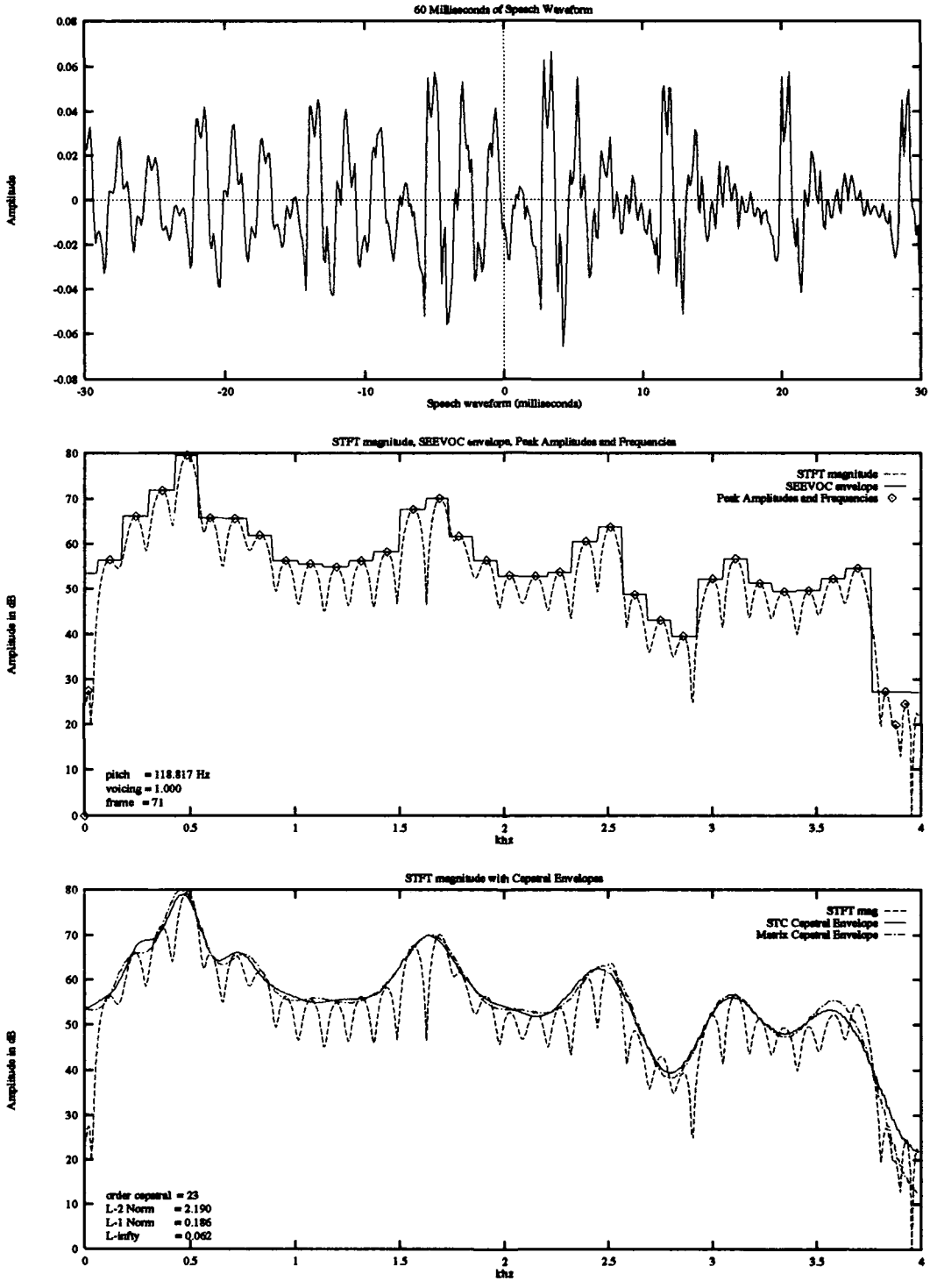


Figure 4.3. Comparison of cepstral envelope generated from an exact matrix computation with frequency warping on male sentence mmcm.si1089 (from TIMIT), and the same cepstral fit generated via STC. The matrix cepstral envelope was computed with 23 coefficients, vice 28 coefficients for the STC envelope.

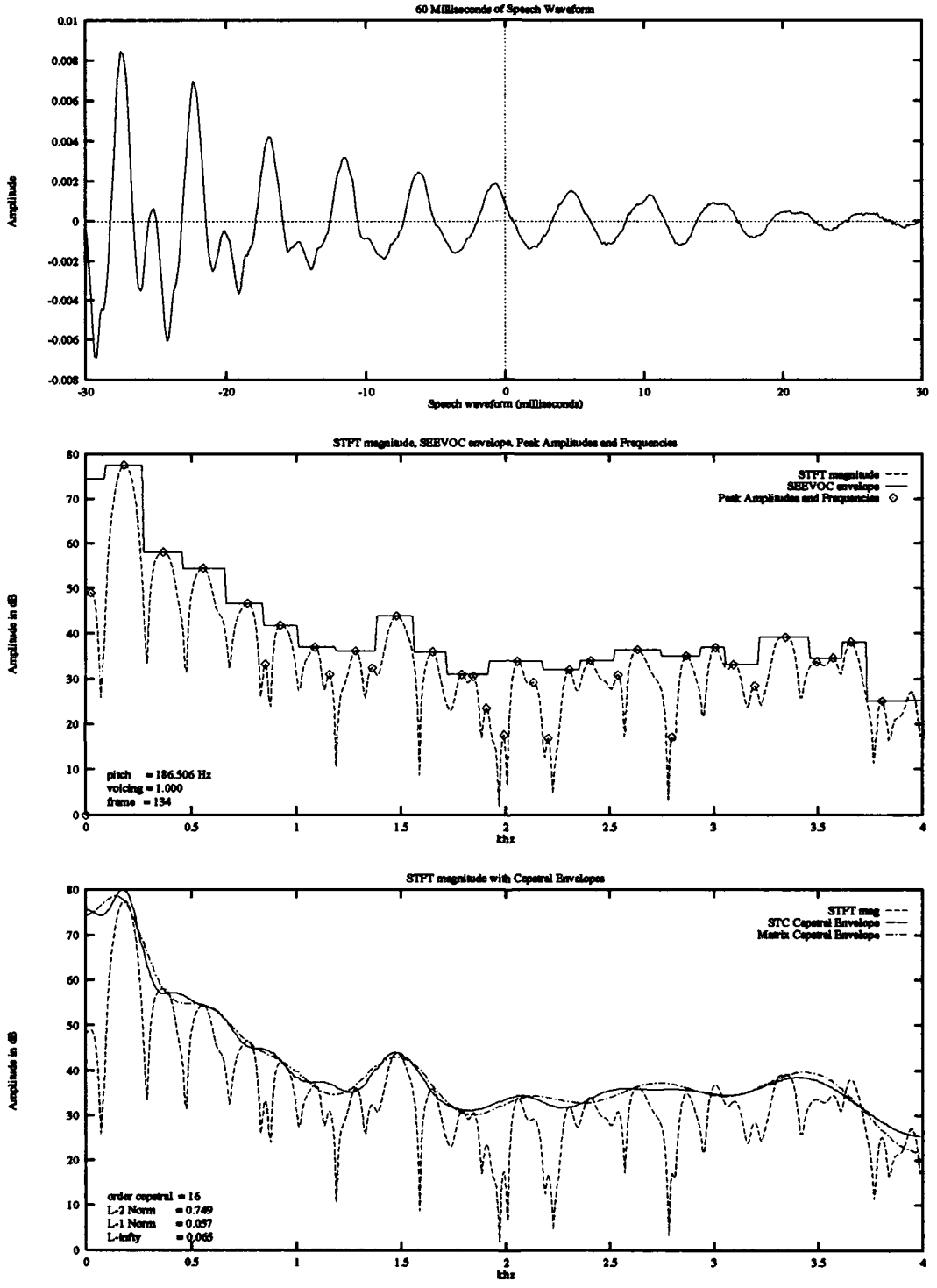


Figure 4.4. Comparison of cepstral envelope generated from an exact matrix computation with frequency warping on female sentence fcmm.si453 (from TIMIT), and the same cepstral fit generated via STC. Only 16 cepstral coefficients were used to compute the matrix cepstral envelope; the STC cepstral envelope was computed with 28 coefficients.

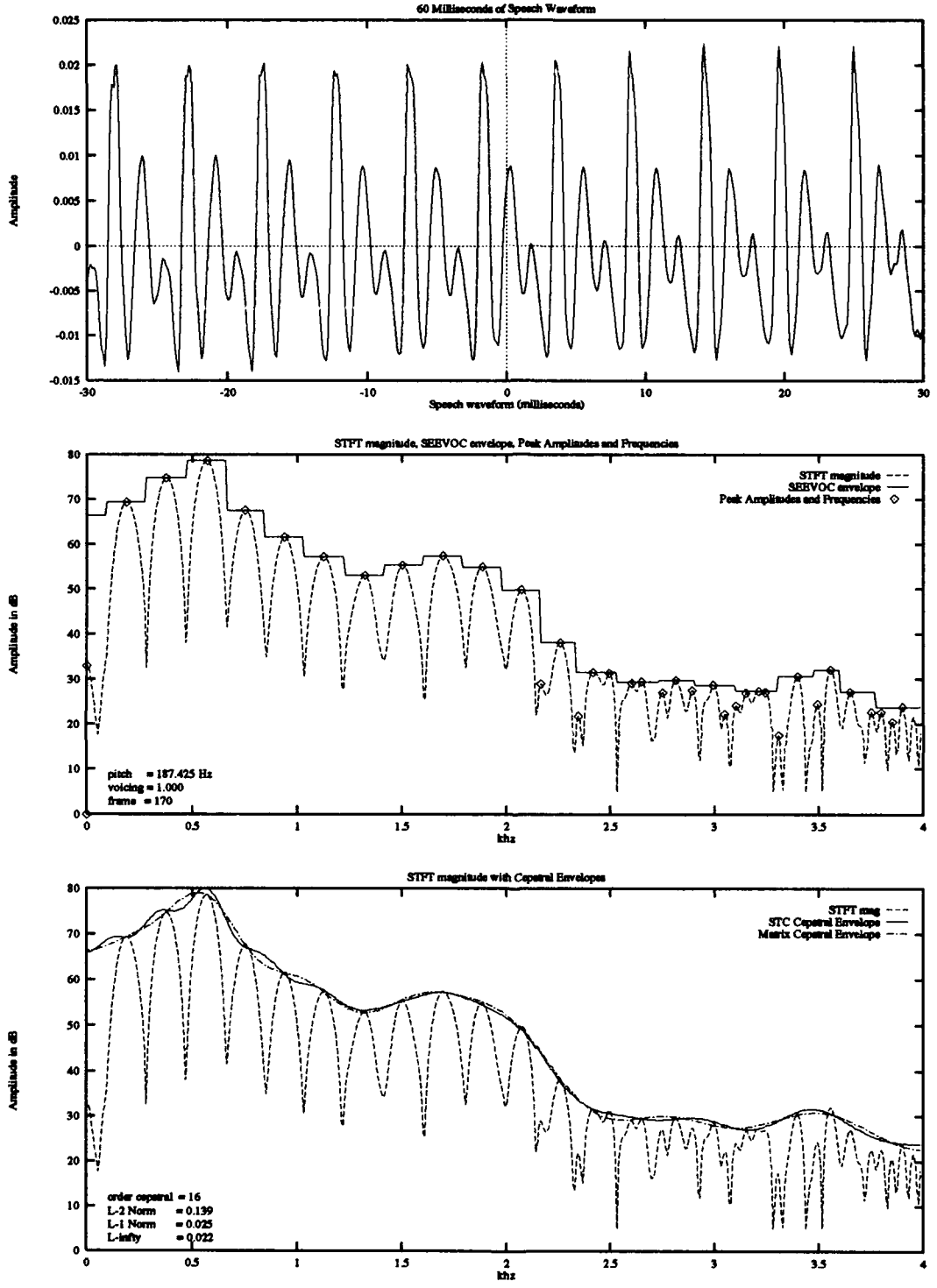


Figure 4.5. Comparison of cepstral envelope generated from an exact matrix computation with frequency warping on female sentence fcmm.si453, and the same cepstral fit generated via STC. Again, only sixteen cepstral coefficients were used to compute the matrix cepstral envelope; the STC cepstral envelope was computed from 28 coefficients.

Toeplitz Approximations. The Toeplitz approximations to the matrix equation $Bc = \gamma$ yield reconstructed speech that is nearly identical in sound to the original speech waveform: an untrained listener may not be able to tell the difference. This is an understandable result since the Toeplitz matrix is not a sparse matrix and the cosine summations within the Toeplitz form approximate those in the B matrix, without setting the summations of the diagonal equal to zero, as is done with the diagonal approximations.

Three Toeplitz approximations were considered: a symmetric form, a nonsymmetric form, and a form constructed by averaging the values over the entire B matrix. Since the latter form requires over twice the computational complexity as the former forms (and since informal listening tests did not indicate any improvement in the synthesized speech using the latter Toeplitz form), this form is not deemed a candidate for real-time implementation.

Informal listening tests of speech synthesized using the symmetric and nonsymmetric Toeplitz approximations to $Bc = \gamma$ to compute the cepstral coefficients indicate the symmetric Toeplitz matrix seems to offer a better reconstruction of the original speech.

The next set of plots compare the STC cepstral envelope with the cepstral envelopes generated from the symmetric and nonsymmetric Toeplitz systems.

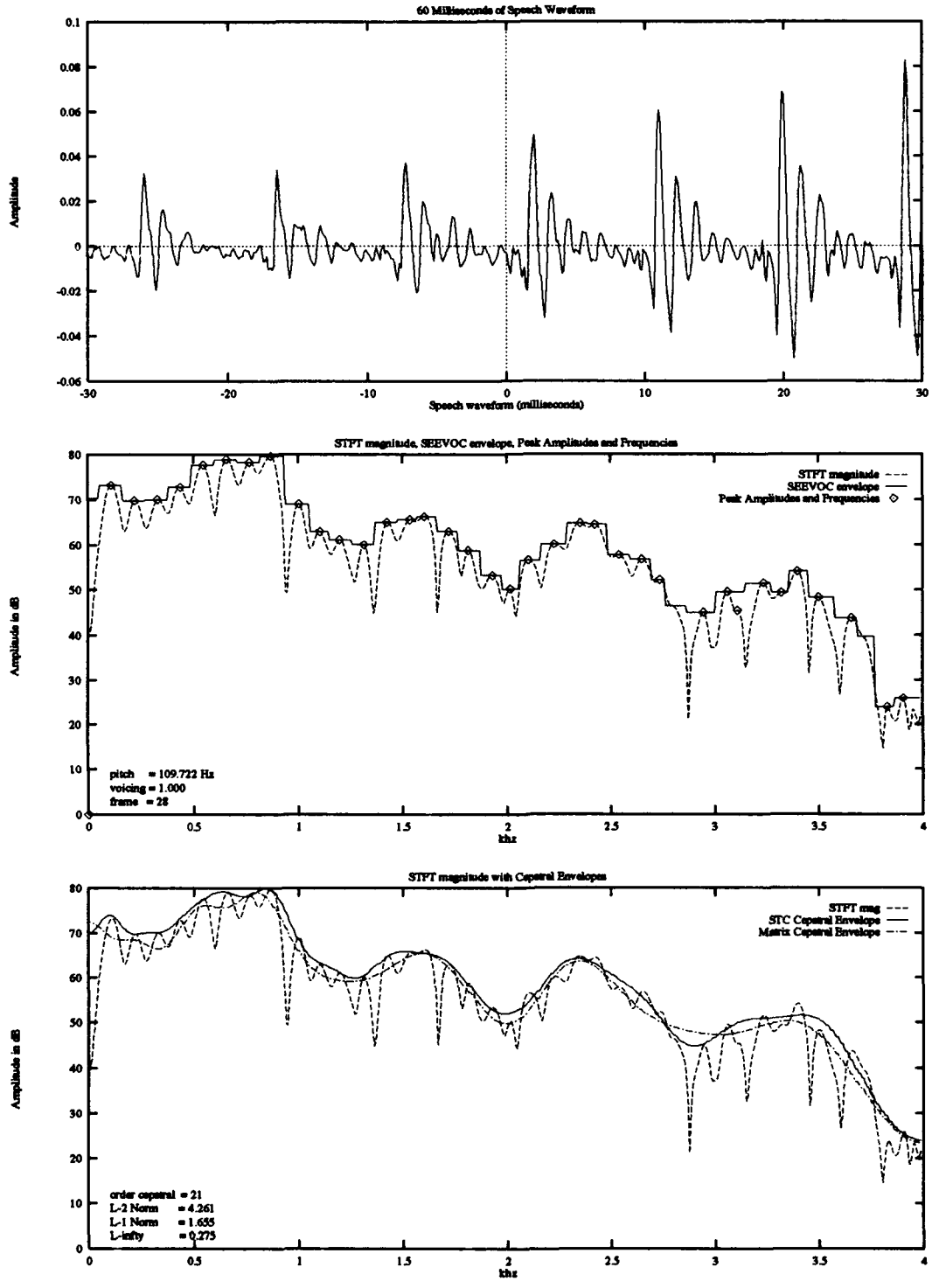


Figure 4.6. Comparison of cepstral envelope generated from a symmetric Toeplitz matrix computation with frequency warping on male sentence mmcm.si1089, and the same cepstral fit generated via STC. The L-2, L-1, and L-infinity norms measure slightly higher distances between these cepstral envelopes than they do between the STC and exact matrix cepstral envelopes.

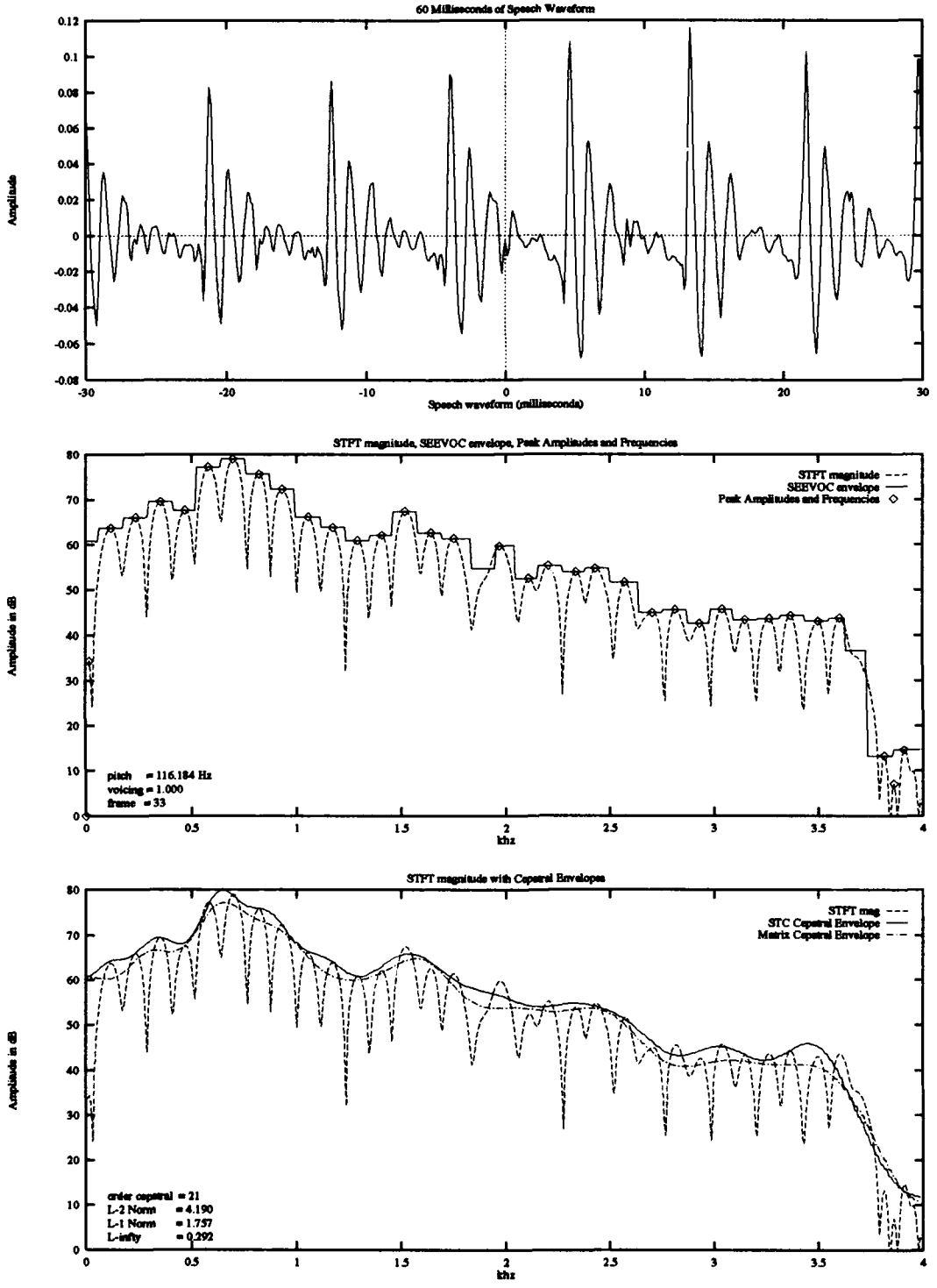


Figure 4.7. Comparison of cepstral envelope generated from a symmetric Toeplitz matrix computation with frequency warping on male sentence mmcm.si1089, and the same cepstral fit generated via STC.

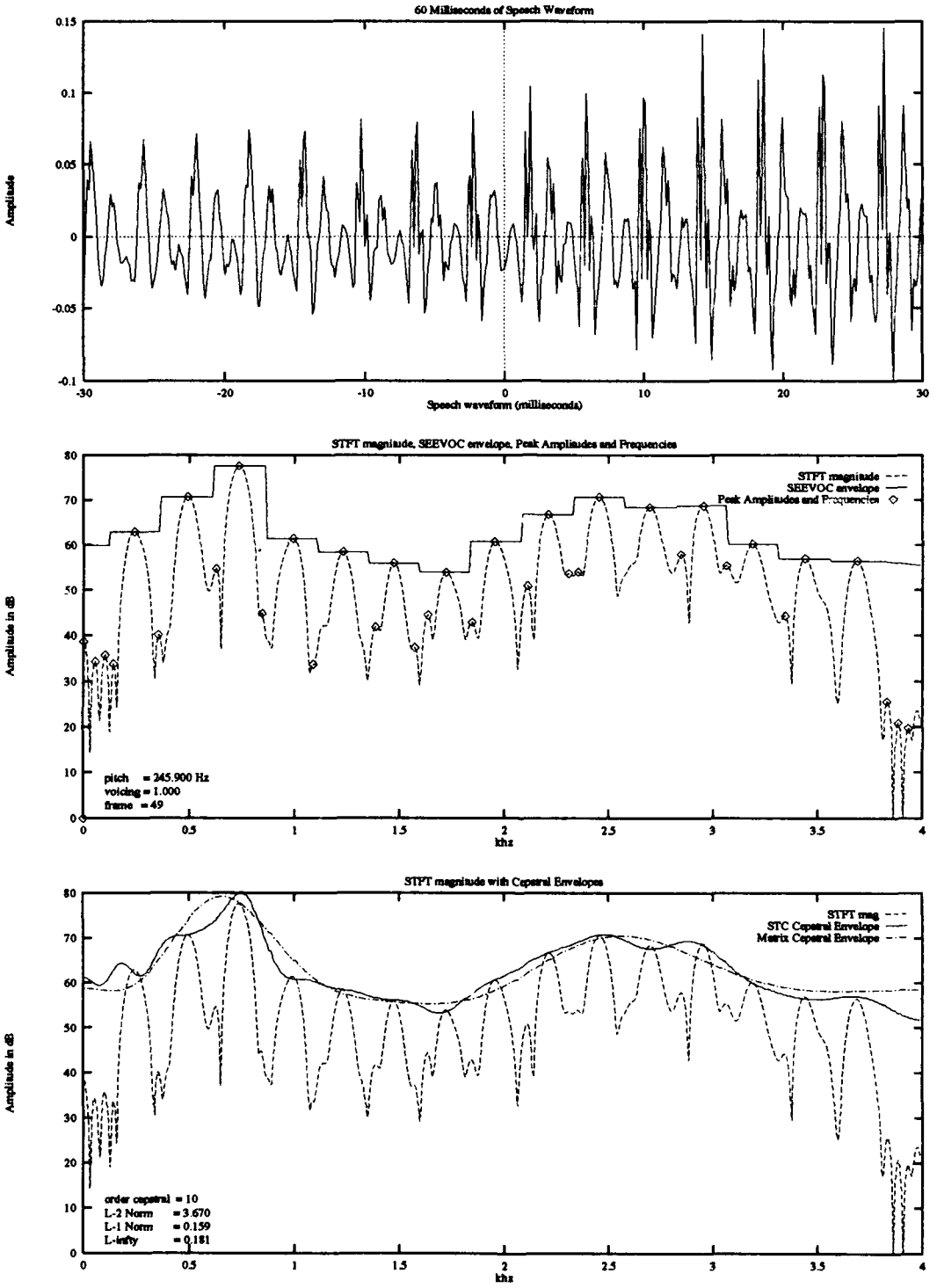


Figure 4.8. Comparison of cepstral envelope generated from a nonsymmetric Toeplitz matrix computation with frequency warping on female sentence fcmm.si453, and the same cepstral fit generated via STC. Only 10 cepstral coefficients were used to compute the matrix cepstral envelope, vice 28 for the STC cepstral envelope.

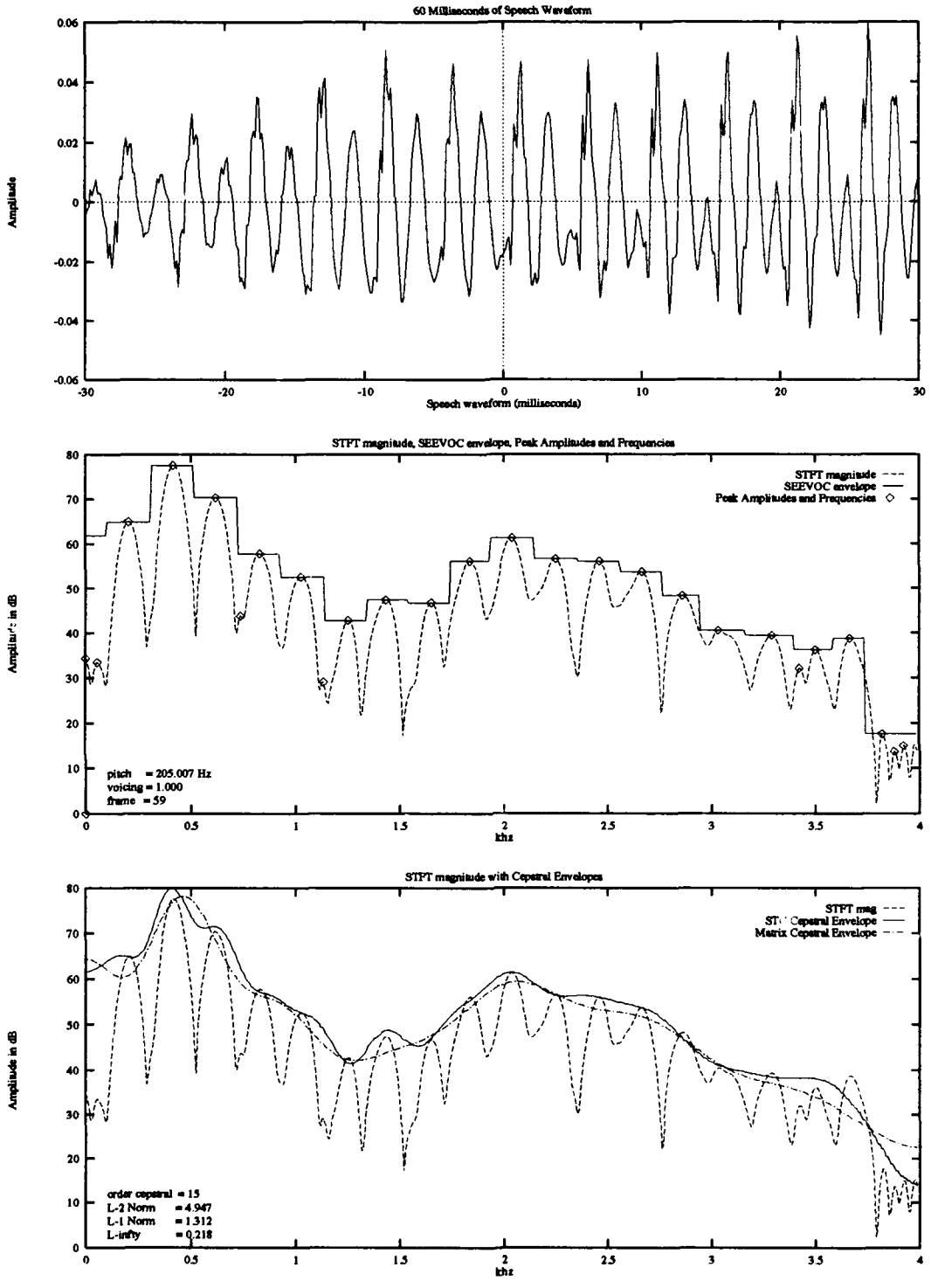


Figure 4.9. Comparison of cepstral envelope generated from a symmetric Toeplitz matrix computation with frequency warping on female sentence fcrh.sa1, and the same cepstral fit generated via STC. The Toeplitz approximation to $Bc = \gamma$ was solved here using the lookahead Levinson algorithm (no instabilites within the matrix were encountered). Only 15 cepstral coefficients were used to compute the matrix cepstral envelope, vice 28 for the STC cepstral envelope.

Diagonal Approximations The tridiagonal and identity approximations to the matrix equation $Bc = \gamma$ yield speech that is understandable but that clearly does not sound identical to the original digitized speech signal. These results are not surprising as both of these forms severely alter the original B matrix by approximating the cosine summations off of the diagonal entries to zero. In the identity case, a matrix entries are not even computed: the cepstral coefficient c_m is equivalent to γ_m.

As explained in the previous chapter, there is valid reasoning behind these approximations: a mathematical analysis shows the cosine summations off the diagonal should approach zero (at least for perfectly voiced speech). But, by setting them all equal to zero and solving for the cepstral coefficients, enough sine wave frequency information is lost to cause the synthesized speech to be clearly distorted from the original.

The next set of plots compare the cepstral envelope as computed within STC using the procedure described in section 2.4.1 with the cepstral envelope generated by a tridiagonal approximation to $Bc = \gamma$. The differences in the two envelopes visually explain the distortions present in speech reconstructed using the tridiagonal matrix system.

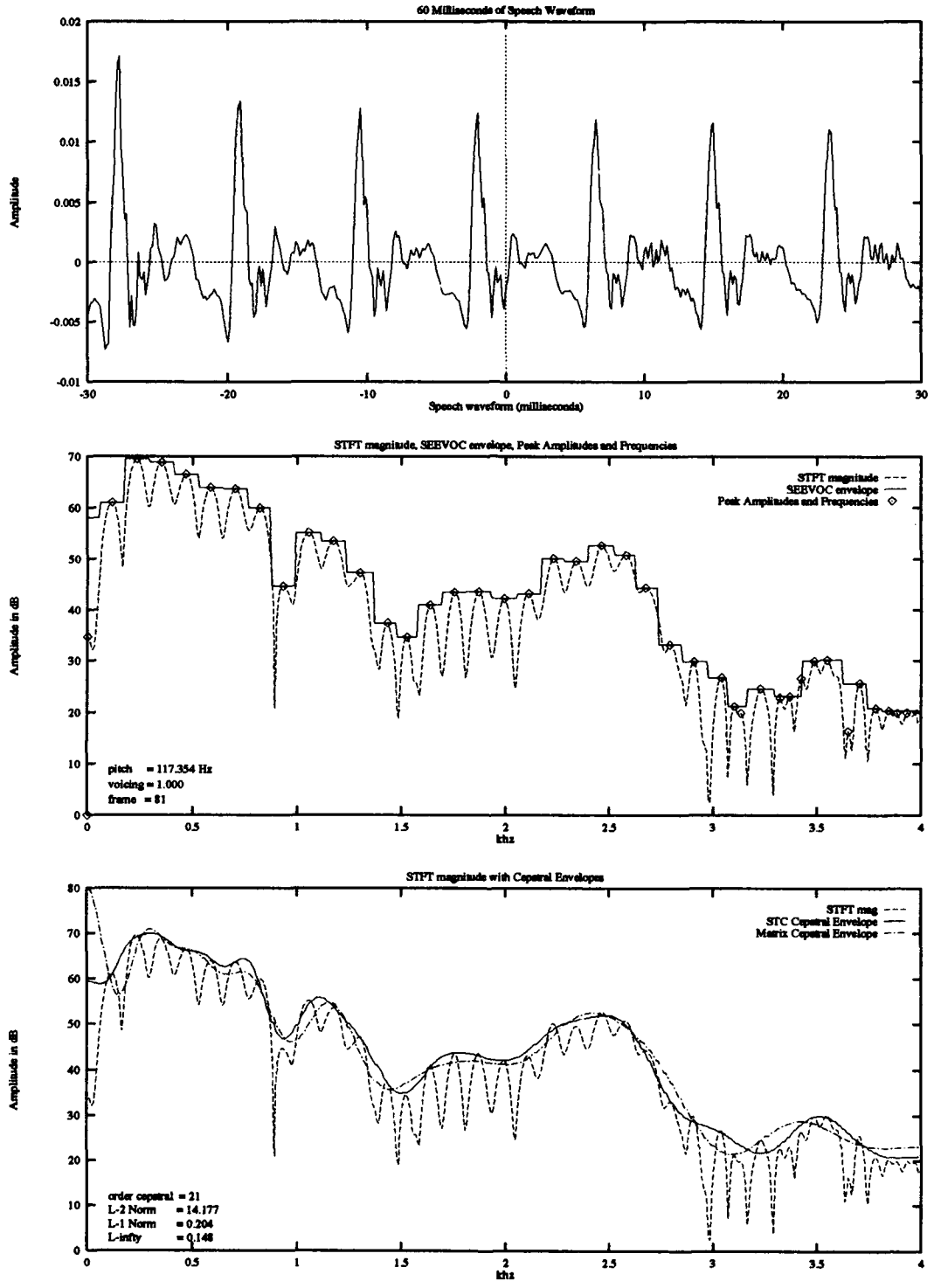


Figure 4.10. Comparison of cepstral envelope generated from a tridiagonal matrix computation with frequency warping on male sentence mmcm.si1089, and the same cepstral fit generated via STC. The L-2 norm measures much higher distances between these cepstral envelopes than they do between the STC and exact matrix cepstral envelopes and between the STC and Toeplitz matrix cepstral envelopes. This result is expected as the tridiagonal approximation is a sparse approximation to the B matrix.

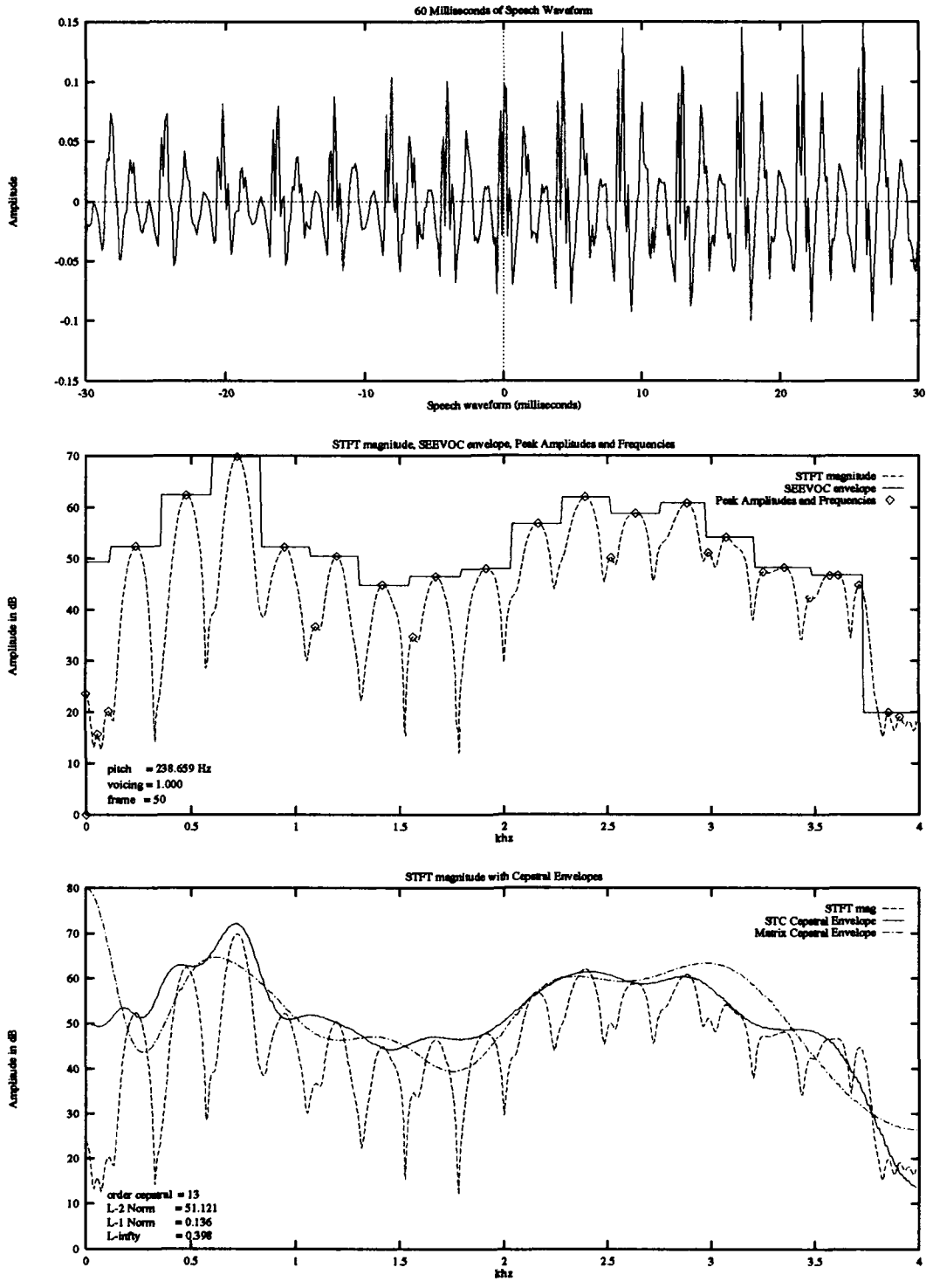


Figure 4.11. Comparison of cepstral envelope generated from a tridiagonal matrix computation with frequency warping on female sentence fedw.sa1, and the same cepstral fit generated via STC. Only 13 cepstral coefficients were used to compute the matrix cepstral envelope, vice 28 coefficients for the STC cepstral envelope.

The next set of plots illustrate the differences between the STC cepstral envelope and the identity approximation to $Bc = \gamma$. As with the tridiagonal case, the differences visually explain the distortions present in the speech reconstructed using the identity system.

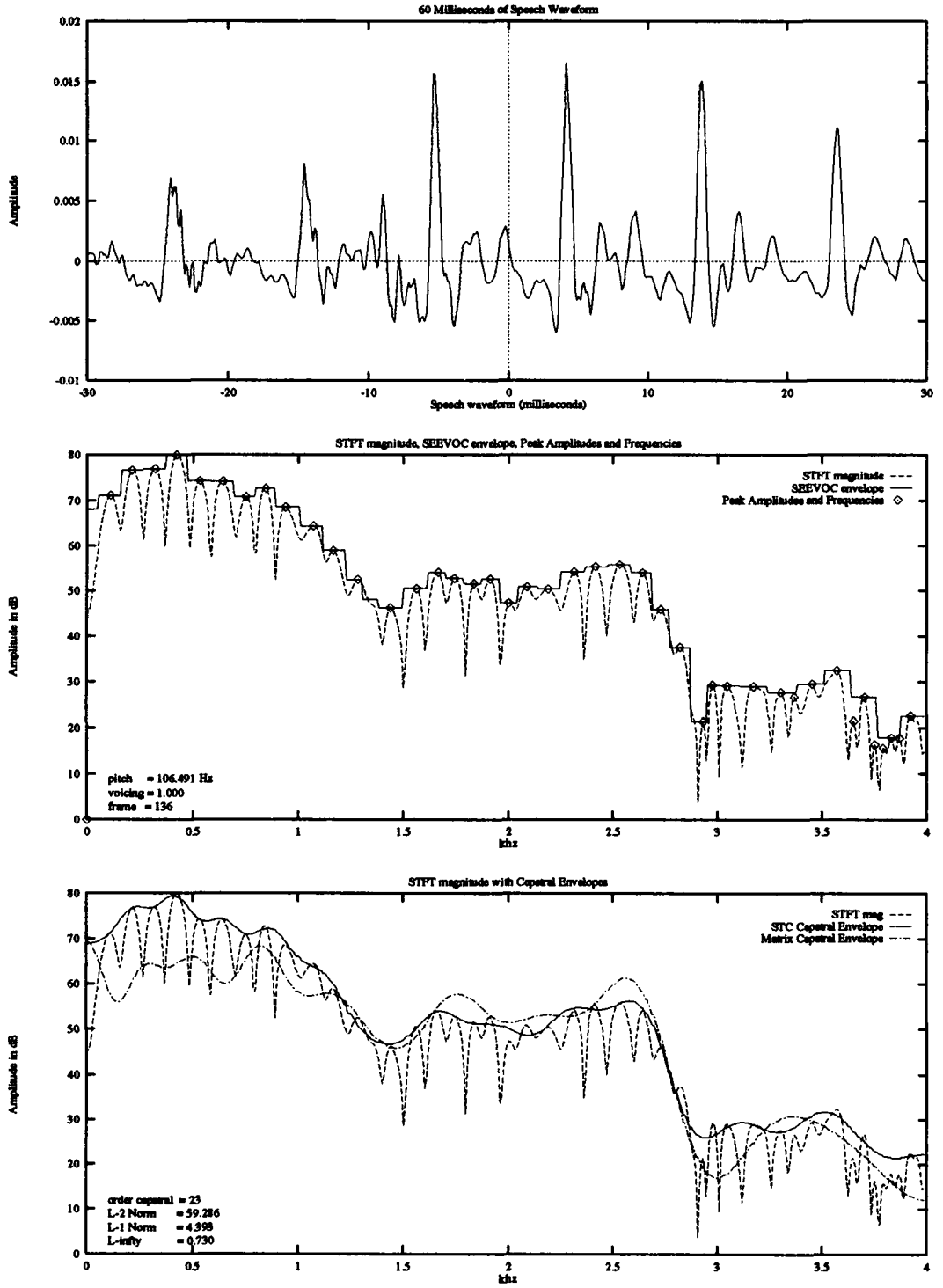


Figure 4.12. Comparison of cepstral envelope generated from an identity matrix computation with frequency warping on male sentence mmcm.si1089, and the same cepstral fit generated via STC.

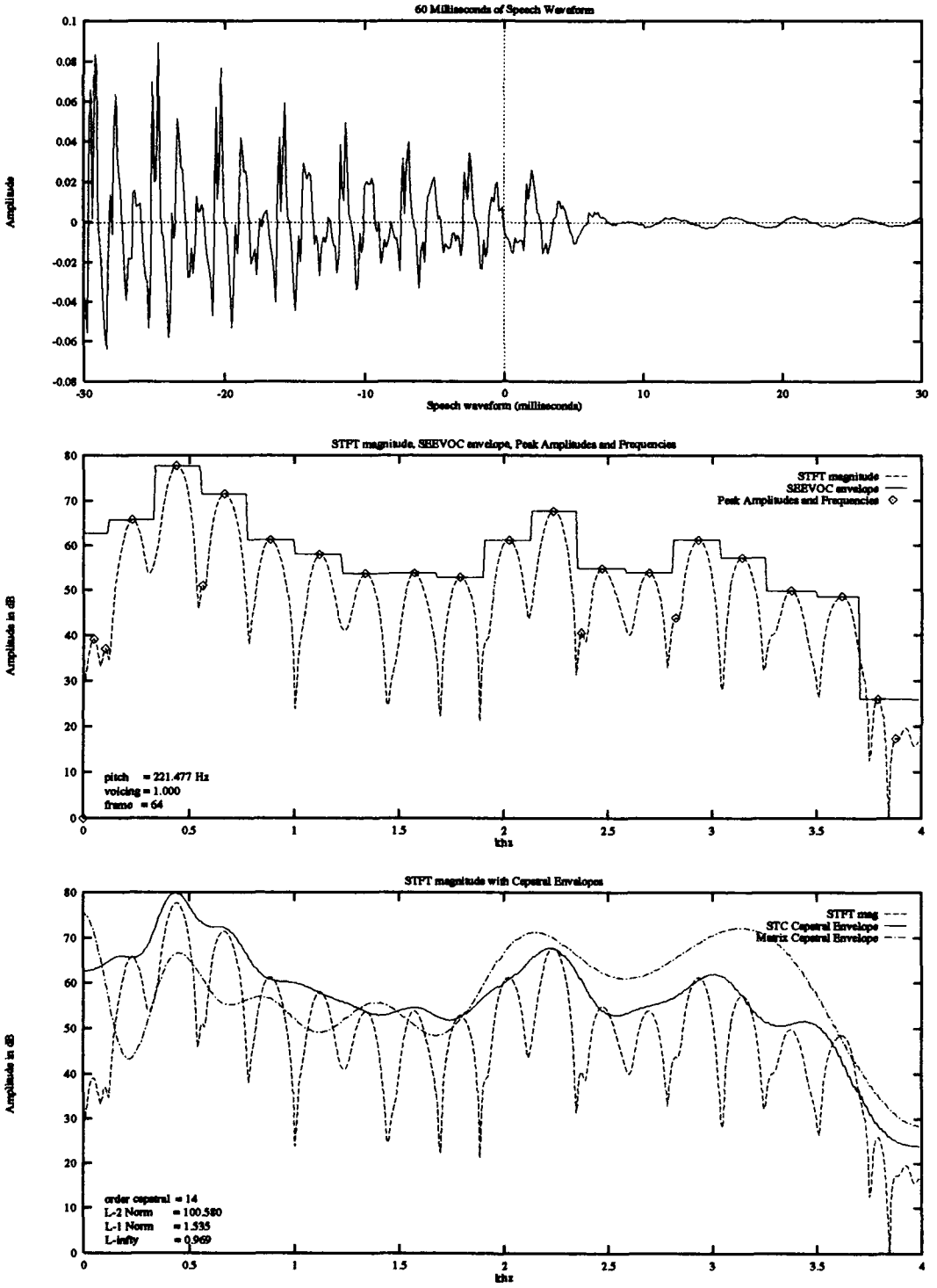


Figure 4.13. Comparison of cepstral envelope generated from an identity matrix computation with frequency warping on female sentence fedw.sa1, and the same cepstral fit generated via STC.

Even though distortions are present in the reconstructed speech using these diagonal approximations, the speech is clearly understandable and not unpleasant sounding, and therefore these approximations may be useful within real-time speech coding systems which require that the reconstructed signal only be intelligible, and not perceptually equivalent to the orignal speech. These algorithms are the most computationally efficient of those discussed.

Unvoiced Speech. As mentioned in Chapter 3, the algorithm developed under this thesis is based on the assumption that *voiced* speech is being processed, thereby resulting in a series of harmonic underlying sine waves for the speech waveform. The harmonic properties of the sine waves ensure that the STFT magnitude is being sampled at equally spaced intervals during the peak finding algorithm. However, in order for the algorithm to be valuable, it must be sufficient to process *unvoiced* speech as well. Below are cepstral plots of an unvoiced speech segment demonstrating that the algorithm is sufficient even when the underlying sine waves are aharmonic. Informal listening tests and the spectrograms of Appendix A also verify the algorithm's sufficiency for processing unvoiced speech.

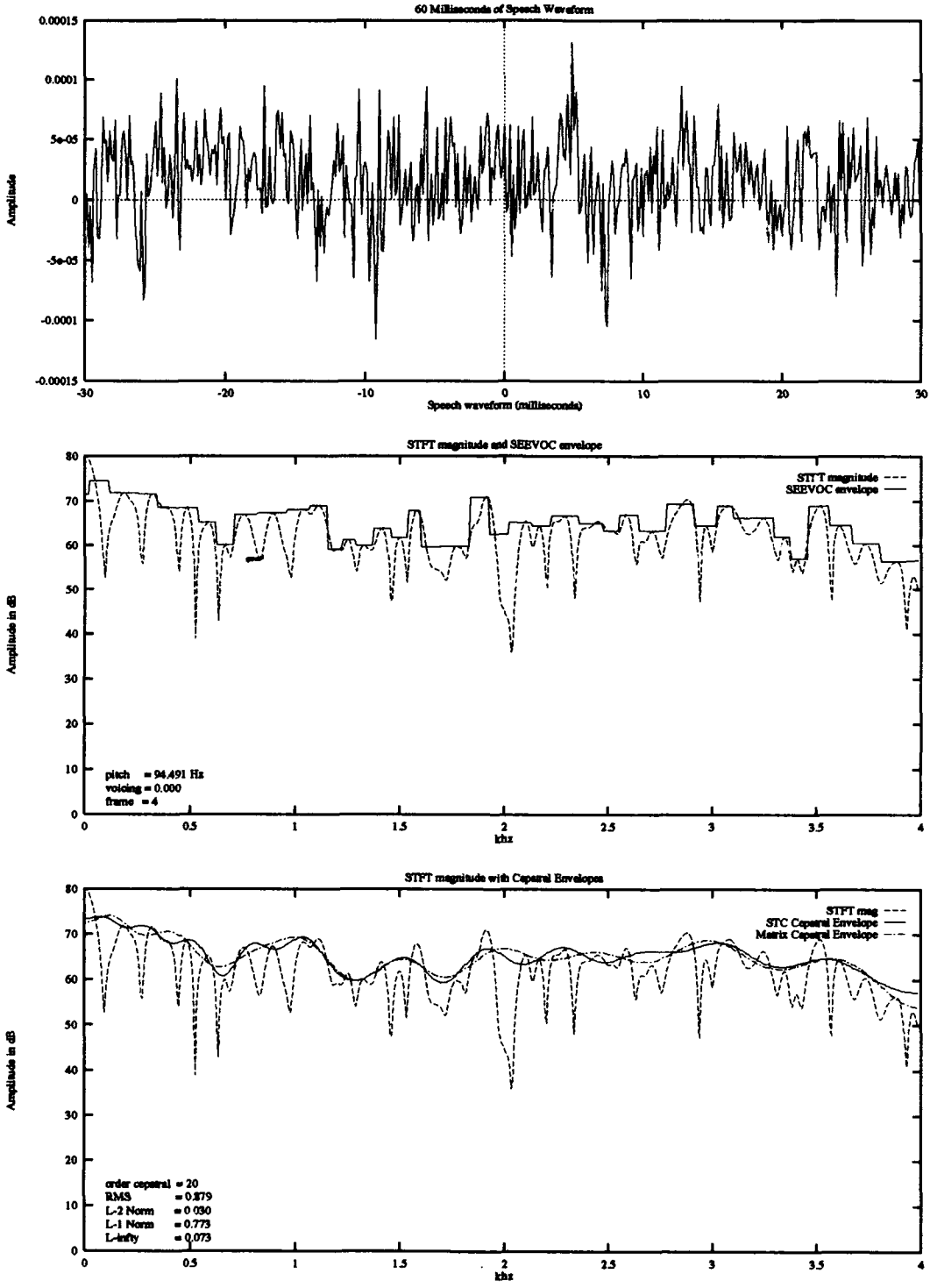


Figure 4.14. Cepstral envelope comparisons of the STC envelope with an envelope generated via the exact matrix computation of $Bc = \gamma$, with frequency warping on male sentence mmcm.si1089 during a period of unvoiced speech.

Summary

Based on both visual and audio results of the methodology presented in the previous chapter, the solution for the cepstral coefficients based on fitting the underlying sine wave amplitudes and frequencies to a cepstral model is a valid solution. Furthermore, the form of the solution (a set of simultaneous linear equations denoted as $Bc = \gamma$) has nice properties which allow it to be approximated with less computationally expensive solutions. Three approximations were investigated: Toeplitz, tridiagonal, and identity forms. The Toeplitz forms offer reconstruction of the original speech comparable to that of the exact solution, with a much lower computational cost. Since Toeplitz systems occur often in DSP applications, and DSP applications are frequently implemented as real-time processes, fast solutions to Toeplitz systems have been developed (1) (18).

The method currently used within STC to compute the cepstral coefficients is correct in its computational accuracy. The value of the algorithm developed under this thesis is therefore not only its computational accuracy but its computational efficiency as well: the algorithm must not only be valid, but must be more computationally efficient than the current STC algorithm, in order to improve STC's real-time implementation.

As mentioned in Chapter 2, the computationally efficiency of the current STC algorithm is of $O(Y) + O(MY)$ where Y is the length of the discrete STFT magnitude spectrum and M is the number of cepstral coefficients. For a 512-point spectrum envelope represented by 28 cepstral coefficients, the computationally complexity is approximately 14,800 FLOPS per cepstral envelope.

For the Toeplitz approximation under this thesis, the computational complexity is $O(M^2)$ for the solution to $Bc = \gamma$ and $O(MK)$ for both the building of the Toeplitz matrix and the building of the γ vector, for a total computational complexity of $O(2MK) + O(M^2)$. Under this algorithm, the number of cepstral coefficients computed (M) can vary between 10 and 28 coefficients for each inner 10 millisecond speech frame, depending on the speaker's pitch. The number of sine waves (K) for each speech frame usually varies between 15 and 60. Therefore, the computational

complexity lies approximately between 400 and 4,200 FLOPS per cepstral envelope. A tridiagonal approximation to $Bc = \gamma$ is of $O(3MK + M)$: $O(2MK)$ for the contruction of B; $O(MK)$ for the construction of γ; and $O(M)$ for the system solution. A tridiagonal approximation requires approximately between 460 and 5,000 FLOPS per cepstral envelope, while an identity solution (involving only the computation of γ) requires approximately between 150 and 1700 FLOPS per envelope. The evaluations of the cosine operations and logarithmic functions are not taken into account in computing the number of FLOPS. It is assumed cosine look-up tables such as those present within STC are used for the cosine operations. The necessary logarithmic function is approximated in both instances by a method more efficient for real-time implementation and already in-place within STC.

The FLOP counts of both the current STC implementation and the approximations of $Bc = \gamma$ indicate a significant computational savings for the Toeplitz, tridiagonal, and identity approximations to $Bc = \gamma$.

Speech synthesized using the tridiagonal and identity approximations is clearly not identical to the original digitized speech signal, but the speech is understandable and may therefore prove useful for real-time environments that do not require speaker recognition or identification. However, speech synthesized using the Toeplitz approximation is nearly identical to the original speech (with only minimal speech degradation, usually during *unvoiced* speech segments) and is therefore a valid candidate for implementation within a multi-purpose real-time speech coding environment.

V. Conclusions and Recommendations.

Research Question One Conclusions

A correct solution to the cepstral coefficients based on fitting a cepstral model to the measured speech data can be developed. The derivation is developed in Chapter 3 and the results presented in Chapter 4. The solution is based on a mathematical analysis using proven and widely accepted DSP ideas and techniques, including Oppenheim and Schafer's complex cepstral model and the mean- and sum-squared error criterions. Basic laws of calculus are also employed. The result is a simultaneous set of linear equations whose solution is the cepstral coefficients for the measured speech data.

This solution is particularly nice as matrices are very common in DSP and other mathematical and scientific applications, and efficient algorithms have been developed to solve matrix equations, like those arising within this thesis solution.

Research Question Two Conclusions

Three mathematical approximations to the solution for the cepstral coefficients are realized.

The tridiagonal and identity approximations are found to be the fastest and most efficient approximations of the B matrix. When using these approximations to solve for the cepstral coefficients within STC, the reconstructed speech, although understandable, is not perceptually equivalent to the original speech. However, these computationally efficient algorithms may prove useful in a speech coding environment that does not require speaker recognition or identification.

As mentioned in the previous chapter, the Toeplitz approximation to $Bc = \gamma$ yields the most promising cepstral coefficients at a reasonable computational cost. Two forms of the Toeplitz matrix are possible for implementation within a real-time environment: a symmetric matrix and a nonsymmetric matrix. Both of these are solvable using the Levinson algorithm found in (18:54-58), and versions of the lookahead Levinson algorithm discussed in (5). The lookahead Levinson

algorithm is a more sophisticated algorithm than the general Levinson algorithm, and is able to solve a large class of Toeplitz systems, at a higher computational cost if any ill-conditioned submatrices occur. However, no ill-conditioned submatrices have been encountered in experiments conducted to date using Toeplitz approximations to $Bc = \gamma$.

Research Question Three Conclusions

The results of the implementation of the derived solution and its approximations are promising. For the exact solution and its Toeplitz approximation, reconstructed speech sounds nearly identical to the original speech (with only minimal speech degradation, usually during *unvoiced* speech segments). Since the Toeplitz solution is an order of magnitude faster than the exact solution, it is the best candidate for real-time implementation within a multi-purpose speech-coding environment. The Toeplitz solution is also less computationally complex than the solution currently implemented within STC. The tridiagonal and identity solutions may prove useful within a real-time speech coding environment that only requires intelligibility and not perceptual equivalence to the original speech signal.

Summary

Progress within speech processing technology, including speech coding, directly supports the military applications described at the beginning of this thesis (22). In particular, STC (developed under Air Force sponsorship) is applicable in such military requirements as security, digital data transmission, the processing of distorted speech, the processing of other sounds such as music and underwater noises, and various narrow-band communications requirements (8-13). Military speech requirements are most often required to be completed within real-time.

This thesis considered an algorithm to speed the computation of cepstral coefficients for speech coding. The algorithm was successfully developed based on current speech coding practices and

the utilization of proven DSP techniques, and implemented within an existing operational speech coding system (STC). Due to the low computational complexity, the algorithm's approximations are definite candidates for implementation within real-time speech coding systems. The Toeplitz approximation to the algorithm offers a unique advantage to speech coding systems: it is a computationally efficient algorithm which can maintain a good quality of speech at the same (or perhaps lower) bit rate.

Iterative improvements to the solution for the cepstral coefficients derived within this thesis can continue to be made, such as optimizing the algorithm for faster computational throughput as computer technology advances, and optimizing the results of the approximations for specific applications.

Appendix A.

This Appendix contains wide-band spectrograms of speech files synthesized using the McAulay-Quatieri Sinusoidal Transform Coder (STC), with the cepstral coefficients computed using the methods developed in this thesis.

Spectrograms offer a graphic view of the speech signal's frequency content versus time, with frequency on the vertical axis and time on the horizontal axis. The intensity of a frequency component at a particular time within the speech signal corresponds to the darkness of the corresponding spot within the spectrogram (16:100).

Within the spectrograms, the spacing of the vertical striations give an indication of the speakers pitch. The vertical striations are closer together for higher-pitched speakers than for lower-pitched speakers (14:726) (16:101) . It is seen from the spectrograms here that a speaker's pitch varies throughout an utterance, which explains why a variable order cepstral is needed within the cepstral algorithms developed under this thesis.

The dark horizontal bars that appear in the spectrograms denote the resonance frequencies of the vocal tract, which also vary with time throughout a speaker's utterance (14:726).

The spectrograms included here offer a visual understanding of the effect of the various computations of the cepstral coefficients on the reconstructed speech. For comparison purposes, spectrograms are also provided for the original speech signal (digitized at 8kHz), and the speech signal as synthesized by STC. The effects of the tridiagonal and identity approximations are readily apparent, as a significant amount of the vocal tract resonances are lost during the coding process when these approximations are used in computing the cepstral coefficients.

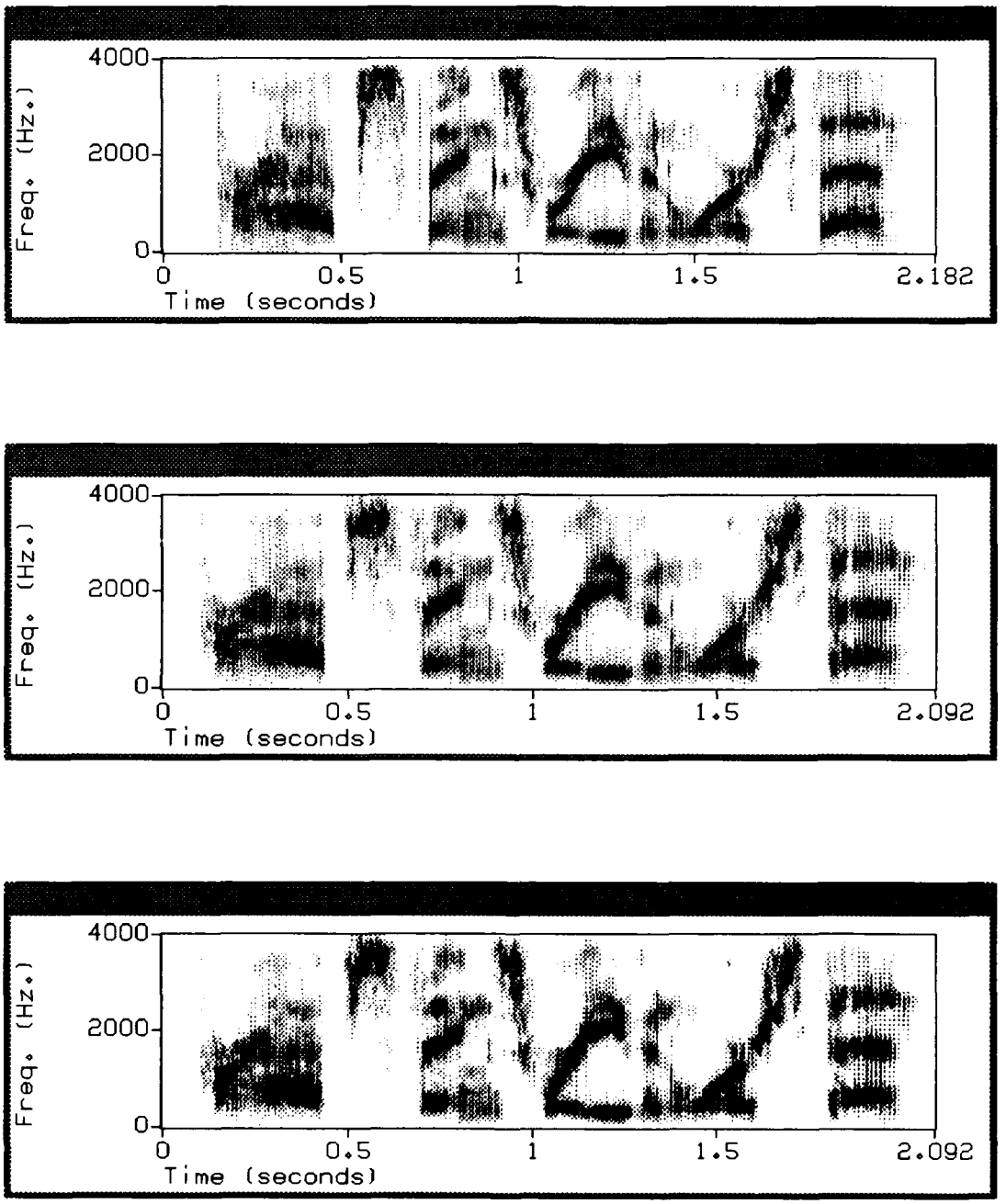


Figure A.1. Spectrograms of the original male utterance mmcm.si1089.8khz, the utterance as synthesized by STC, and the utterance as synthesized within STC with the cepstral coefficients computed using the method of this thesis (via the exact solution to the matrix equation $Bc = \gamma$).

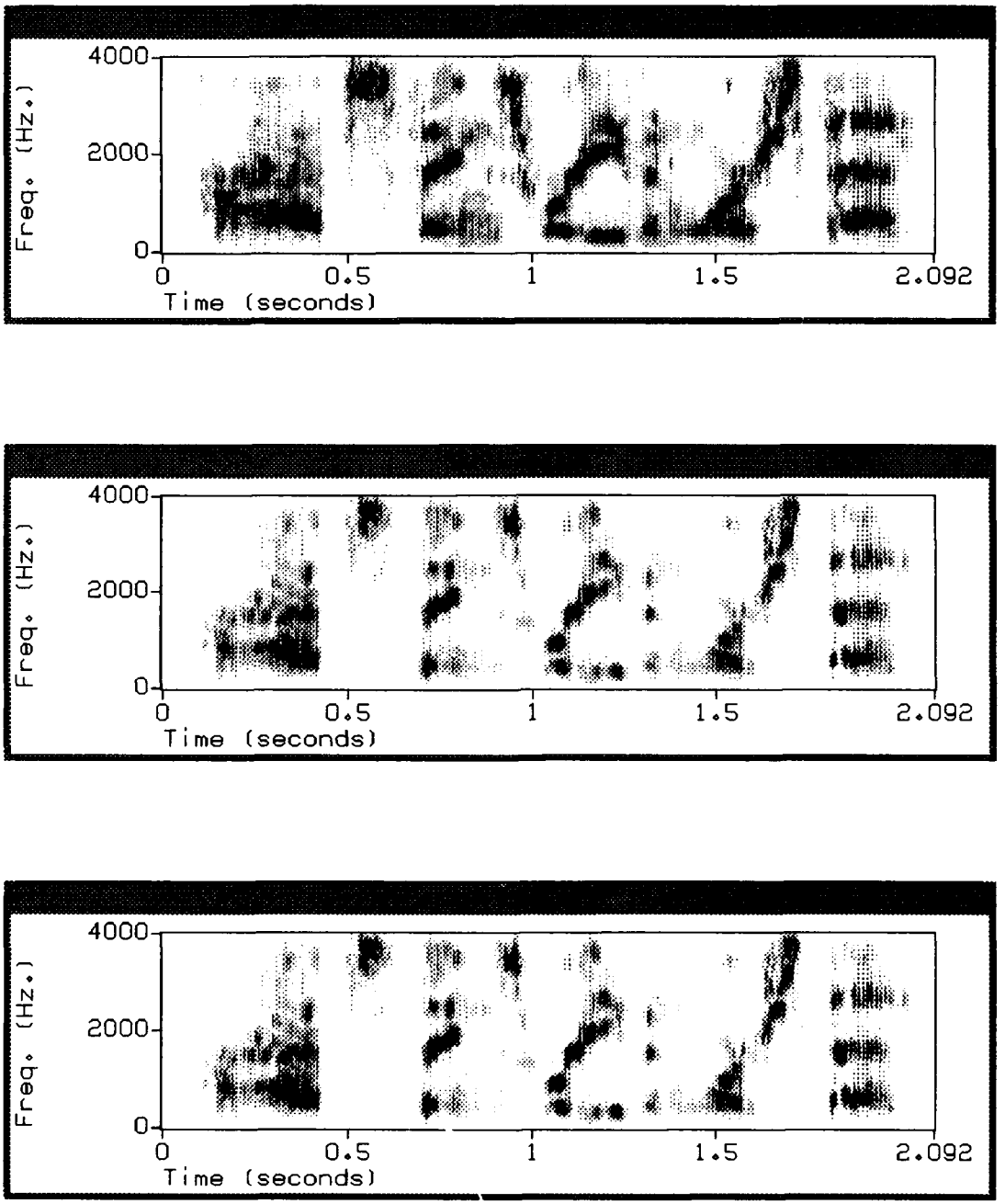


Figure A.2. Spectrograms of the male utterance mmcm.si1089.8khz as synthesized within STC using the Toeplitz, tridiagonal, and identity approximations to the cepstral coefficient equation $Bc = \gamma$.

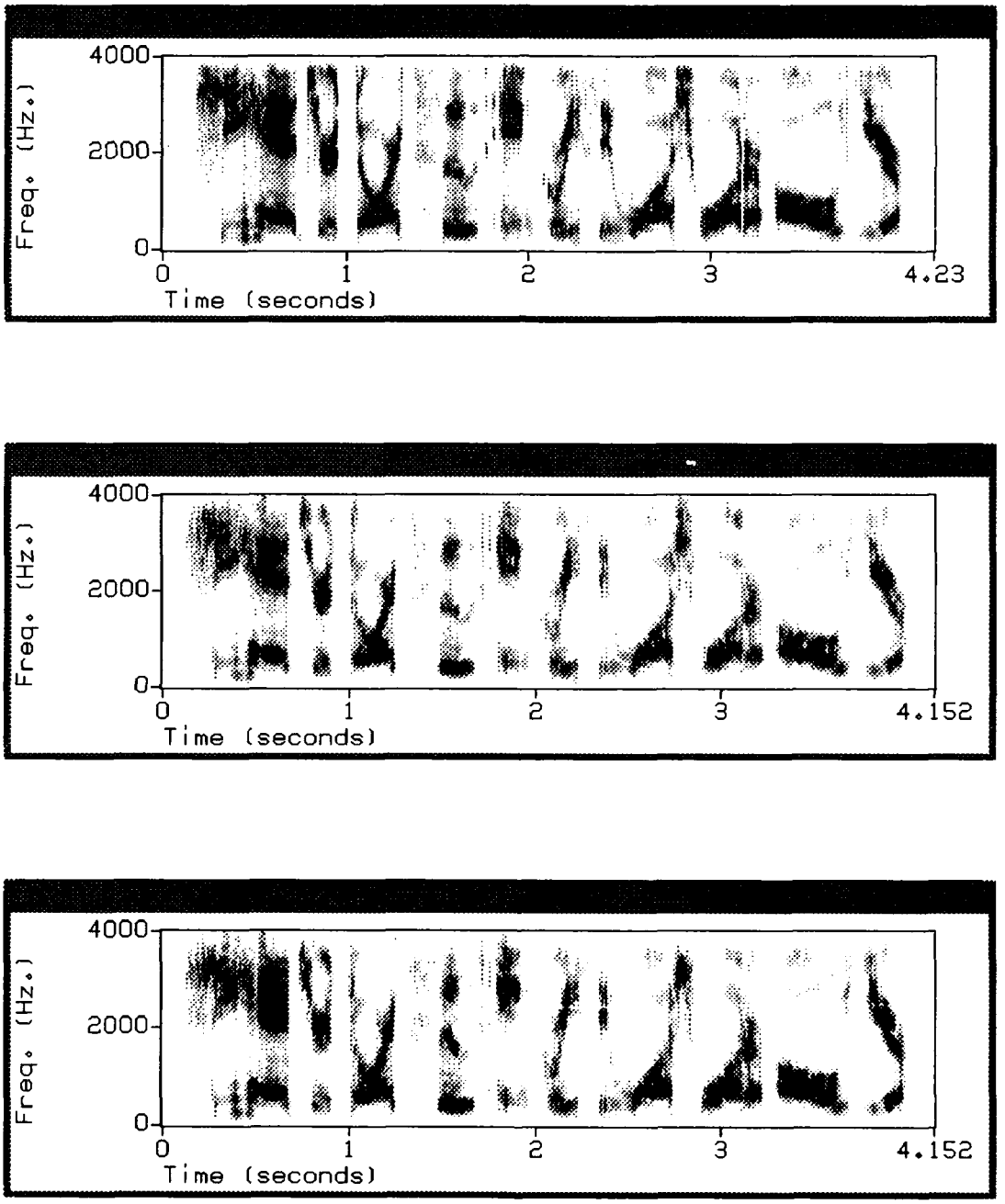


Figure A.3. Spectrograms of the original female utterance fedw.sa1.8khz, the utterance as synthesized by STC, and the utterance as synthesized within STC with the cepstral coefficients computed using the method of this thesis(via the exact solution to the matrix equation $Bc = \gamma$).

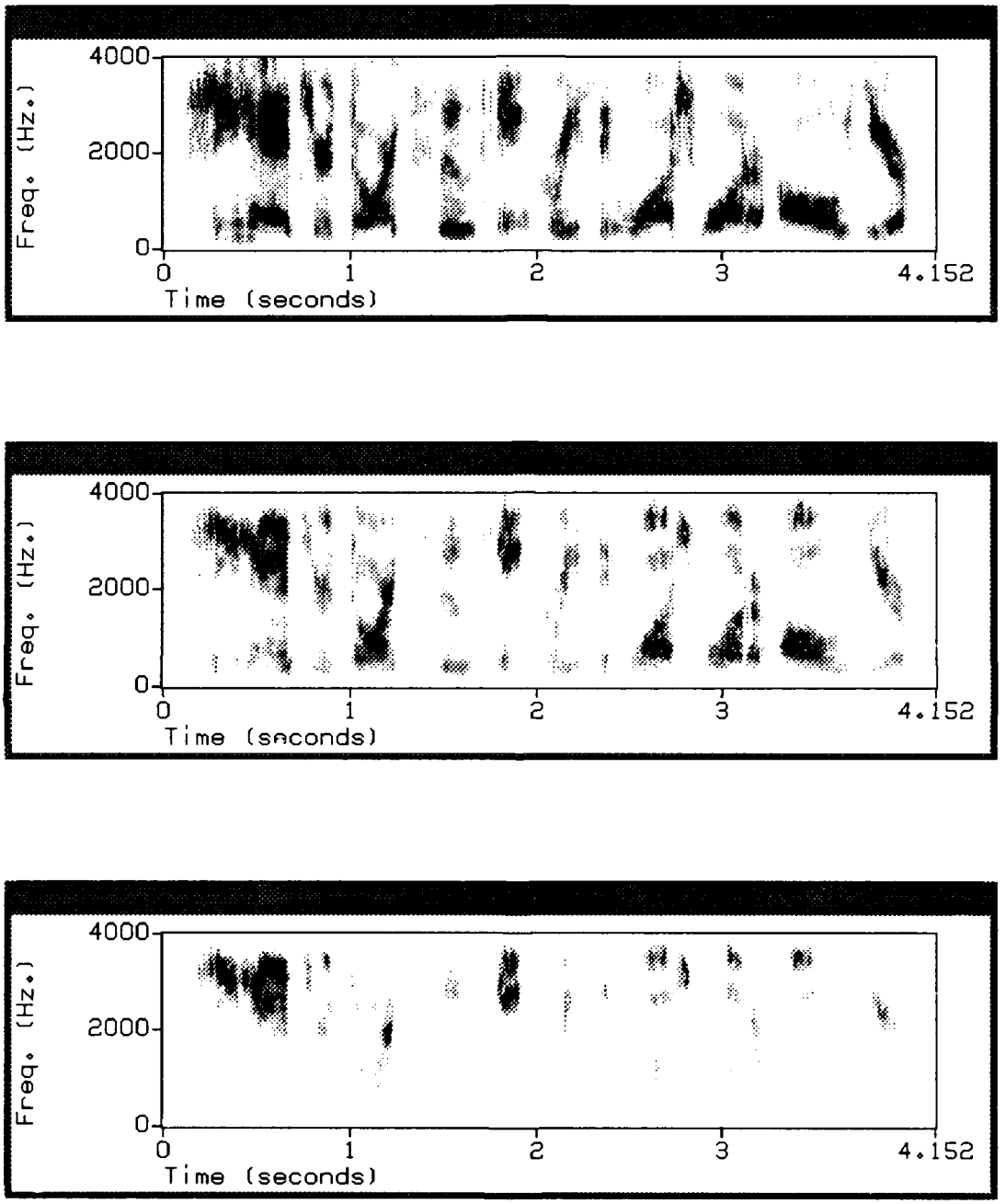


Figure A.4. Spectrograms of the female utterance fedw.sa1.8khz as synthesized within STC using the Toeplitz, tridiagonal, and identity approximations to the cepstral coefficient equation $Bc = \gamma$.

Bibliography

1. Blahut, Richard E. *Fast Algorithms for Digital Signal Processing*. Reading, Massachusetts: Addison-Wesley, 1985.
2. Cormen, Thomas H., et al. *Introduction to Algorithms*. Cambridge: The MIT Press, 1990.
3. Corwin, Lawrence J. and Robert H. Szczarba. *Multivariable Calculus*. New York: Marcel Dekker, Inc., 1982.
4. Golub, Gene H. *Matrix Computations*. Baltimore: John Hopkins University Press, 1983.
5. Hansen, Per Christian and Tony F. Chan. "FORTRAN Subroutines for General Toeplitz Systems," *ACM Transactions on Mathematical Software*, *18*(3) (Sep 1992).
6. Lin, Kun-Shan, et al. "The TMS320C20 Family of Digital Signal Processors," *Proceedings of the IEEE*, *75*(9) (September 1987).
7. McAulay, Robert J., "Personal Correspondance," 24 Sep 92.
8. McAulay, Robert J. and Terrence Champion. "Improved Interoperable 2.4kb/s LPC using Sinusoidal Transform Coder Techniques," *IEEE 1990 International Conference on Acoustics, Speech and Signal Processing*, (S12.1) (1990).
9. McAulay, Robert J. and Thomas F. Quatieri. "Speech Analysis/Synthesis Based on a Sinusoidal Representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *ASSP-34*(4) (August 1986).
10. McAulay, Robert J. and Thomas F. Quatieri. "Computationally Efficient Sine-Wave Synthesis and its Application to Sinusoidal Transform Coding," *IEEE 1988 International Conference on Acoustics, Speech and Signal Processing*, (S9.1) (1988).
11. McAulay, Robert J. and Thomas F. Quatieri. "Phase Coherence in Speech Reconstruction for Enhancement and Coding Applications," *IEEE 1989 International Conference on Acoustics, Speech and Signal Processing*, (S4.23) (1989).
12. McAulay, Robert J. and Thomas F. Quatieri. "Sine-Wave Phase Coding at Low Data Rates," *IEEE 1991 International Conference on Acoustics, Speech and Signal Processing*, *1*(S9.1) (1991).
13. McAulay, Robert J. and Thomas F. Quatieri. *Advances in Speech Signal Processing*, chapter 6, 165–208. Marcel Dekker, Inc., 1992.
14. Oppenheim, Allan V. and Ronald W. Schafer. *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, Inc., 1989.
15. Papamichalis, Panos and Ray Jr. Simar. "The TMS320C30 Floating Point Digital Signal Processor," *IEEE Micro Magazine*, *8*(6) (Dec 1988).
16. Parsons, Thomas. *Voice and Speech Processing*. New York: McGraw-Hill Book Company, 1987.
17. Paul, Douglas B. "The Spectral Envelope Estimation Vocoder," *IEEE Transactions on Acoustics, Speech, and Siganl Processing*, *ASSP-29*(4) (Aug 1981).
18. Press, William H., et al. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge: Cambridge University Press, 1988.
19. Proakis, John G. and Dimitris G. Manolakis. *Introduction to Digital Signal Processing*. New York: Macmillan Publishing Company, 1988.

20. Rabiner, L.R. and R.W. Schafer. *Digital Processing of Speech Signals*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1978.

21. Watkins, David S. *Fundamentals of Matrix Computations*. New York: John Wiley and Sons, 1991.

22. Weinstein, Clifford J. "Opportunities for Advanced Speech Processing in Military Computer-Based Systems," *Proceedings of the IEEE*, *79*(11) (Nov 1991).

Vita

Captain Kimberly A. Walther attended Vanderbilt University, Nashville, Tennessee, graduating in 1986 with a Bachelor of Science degree in Mathematics and Computer Science, and an Air Force commission. After commissioning, she was assigned to the Headquarters Air Force Office of Special Investigations at Bolling AFB, Washington DC. She entered the Air Force Institute of Technology in June 1991.

Permanent address: 1902 Millwood Drive
Conway, Arkansas 72032

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1992	Master's Thesis

4. TITLE AND SUBTITLE

Efficient Derivation and Approximations of Cepstral Coefficients for Speech Coding

5. FUNDING NUMBERS

6. AUTHOR(S)

Kimberly A. Limcangco, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCS/ENG/92D-20

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratories, Hanscom AFB, MA

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

A new formulation is presented for the calculation of cepstral coefficients directly from measured sine wave amplitudes and frequencies of speech waveforms. Approximations to these cepstral coefficients are shown to be suitable for operation in a real-time speech coding environment. These results were encoded in the *C* programming language and then evaluated through experiments that were conducted on the McAulay-Quatieri Sinusoidal Transform Coder (STC).

14. SUBJECT TERMS

Speech Coding, Cepstral Processing

15. NUMBER OF PAGES

90

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL